

**UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN**



**DISEÑO DE UN PÉNDULO INVERTIDO PARA SU USO EN LAS  
ASIGNATURAS DE SISTEMAS DE CONTROL Y CONTROL  
APLICADO DE LA UNIVERSIDAD NACIONAL DE INGENIERÍA  
(UNI)**

Trabajo monográfico para optar al título de:  
Ingeniero Electrónico

Autores:

Br. Félix Ruiz Tórrez

Br. Héctor Andrés Montes Cortéz

Tutor:

Ing. Álvaro Gaitán

Managua, Nicaragua, junio 2019

## **AGRADECIMIENTOS**

A Dios por permitirme culminar esta meta y ponerme los libros y las personas correctas en el tiempo correcto. A mi familia por su apoyo incondicional. A nuestro tutor Ing. Álvaro Gaitán por su guía. A mi amigo Roberto Pomares por su retroalimentación y ánimo brindado en el trayecto de esta tesis. A Gisselle Amaya por su escucha.

Félix Ruiz Tórrez (2019)

Le doy gracias a Dios por haberme acompañado a lo largo de mi carrera. A mis padres por haberme apoyado en todo momento y por todos los valores que me han inculcado, siendo siempre un ejemplo de vida para mí. A mis hermanos por ser parte importante en mi vida. A nuestro tutor, Ing. Álvaro Gaitán por toda su guía. A mis amigos que siempre me han animado a seguir adelante.

Héctor Montes Cortez (2019)

## RESUMEN

En el presente documento se expone el diseño, construcción de un prototipo y funcionalidad de un péndulo invertido, que sirva como herramienta de laboratorio para contribuir al desarrollo de los conocimientos adquiridos por los estudiantes en, las clases de Sistemas de Control y Control Aplicado impartidas en la carrera de Ingeniería Electrónica de la Universidad Nacional de Ingeniería de Nicaragua. La investigación en este trabajo monográfico es de tipo aplicada, porque se utilizó en un sistema inestable la teoría necesaria y los materiales para la elaboración del péndulo invertido, poniendo en práctica los conocimientos adquiridos en el estudio de nuestra carrera de Ingeniería Electrónica.

El péndulo invertido consiste en una varilla atada a un pivote montado sobre un carro e impulsado por un motor. El sistema es inestable ya que la varilla se puede caer en cualquier momento a menos de que se aplique una fuerza de control adecuada a través del giro del motor.

El diseño de un péndulo invertido conlleva mantener la estabilidad de la varilla en posición vertical hacia arriba. Se construyó un prototipo basado en el modelo del péndulo invertido montado en un carro. Se elaboró el modelo matemático identificando sus parámetros. Se diseñaron los controladores Proporcional Integral Derivativo (PID) y Regulador Cuadrático Lineal (LQR). Se utilizó el software matemático Matlab (Matrix Laboratory) permitiendo ejecutar los controladores (PID y LQR) en el sistema real logrando con éxito el funcionamiento y estabilización de la planta del péndulo invertido. Se presentan los programas de Matlab y Simulink utilizados en cada uno de los experimentos realizados. También se elaboró un manual de usuario destinado a dar asistencia técnica a las personas que utilizan esta herramienta.

## **ABSTRACT**

In this document, the design, construction of a prototype and functionality of an inverted pendulum is exposed, which serves as a laboratory tool that contribute to the development of the knowledge acquired by the students in the Applied Control and Control Systems classes taught in the Electronic Engineering major at Universidad Nacional de Ingeniería of Nicaragua. The research in this monographic work is of an applied type, because the necessary theory and the materials for the elaboration of the inverted pendulum were used in an unstable system, putting into practice the knowledge acquired in the study of our Electronic Engineering major.

The inverted pendulum consists of a rod attached to a pivot mounted on a car and driven by a motor. The system is unstable since the rod may fall at any time unless adequate control force is applied through the rotation of the motor.

The design of an inverted pendulum involves maintaining the stability of the rod in an upright vertical position. A prototype was built based on the model of the inverted pendulum mounted on a car. The mathematical model was developed identifying its parameters. The Proportional Integral Derivative (PID) and Linear Quadratic Regulator (LQR) controllers were designed. The mathematical Matlab software (Matrix Laboratory) was used, allowing the controllers (PID and LQR) to be executed in the real system, successfully achieving the operation and stabilization of the Inverted Pendulum plant. The Matlab and Simulink programs used in each of the experiments carried out are presented. A user manual was also prepared to provide technical assistance to people who use this tool.

## ÍNDICE

INTRODUCCIÓN .....	1
JUSTIFICACIÓN .....	2
OBJETIVOS .....	3
OBJETIVO GENERAL .....	3
OBJETIVOS ESPECÍFICOS .....	3
CAPÍTULO 1: MARCO TEÓRICO .....	4
1.1 SISTEMA .....	4
1.2 SOLUCIÓN COMPLEMENTARIA DE EDO LINEALES .....	5
1.3 SISTEMA GENERAL DE SEGUNDO ORDEN .....	7
1.4 SISTEMA DE SEGUNDO ORDEN SUBAMORTIGUADO .....	9
1.5 IDENTIFICACIÓN DE SISTEMAS .....	11
1.5.1 Grey-box Modelling .....	12
1.6 OBTENCIÓN DE UN MODELO A BASE DE DATOS EXPERIMENTALES SEGUN EL MÉTODO DE CUADRADOS MÍNIMOS .....	13
1.7 SISTEMA DE CONTROL .....	14
1.8 SISTEMA DE CONTROL DE LAZO ABIERTO .....	14
1.9 SISTEMA DE CONTROL DE LAZO CERRADO .....	15
1.10 CONTROLADOR PID .....	16
1.11 CONTROLADOR LQR .....	16
1.12 EL PÉNDULO INVERTIDO SOBRE EL CARRO .....	17
1.12.1 Modelado del péndulo invertido .....	19
1.13 EL PÉNDULO SOBRE EL CARRO .....	20
1.14 MODELADO MATEMÁTICO DEL CARRO .....	22

1.14.1 Modelado del motor.....	23
1.15 MODELADO MATEMÁTICO DEL PÉNDULO .....	26
1.16 MODELADO MATEMÁTICO DEL PÉNDULO INVERTIDO.....	28
1.17 ESPACIO DE ESTADOS.....	29
1.18 SCILAB.....	31
1.19 MATLAB .....	32
1.20 SIMULINK.....	32
1.21 ARDUINO .....	33
1.22 CODIFICADOR INCREMENTAL.....	34
1.23 PUENTE H .....	35
CAPÍTULO 2: ANÁLISIS Y PRESENTACIÓN DE RESULTADOS .....	37
2.1 ANÁLISIS DEL TRABAJO .....	37
2.2 CONSTRUCCIÓN DEL PROTOTIPO .....	38
2.3 IDENTIFICACIÓN DE LOS PARÁMETROS DEL PÉNDULO ....	44
2.4 IDENTIFICACIÓN DE LOS PARÁMETROS DEL CARRO.....	50
2.5 IDENTIFICACIÓN DE LOS PARÁMETROS DEL PÉNDULO SOBRE EL CARRO.....	54
2.6 IDENTIFICACIÓN DE LOS PARÁMETROS DEL PÉNDULO INVERTIDO SOBRE EL CARRO. ....	59
2.7 DISEÑO DEL CONTROLADOR PID PARA EL PÉNDULO INVERTIDO SOBRE EL CARRO MEDIANTE EL LUGAR GEOMÉTRICO DE LAS RAÍCES .....	61
2.8 DISEÑO DEL CONTROLADOR LQR PARA EL PÉNDULO INVERTIDO SOBRE EL CARRO .....	75
2.9 IMPLEMENTACIÓN DE LOS CONTROLADORES PID Y LQR EN EL SISTEMA FÍSICO REAL .....	77

CONCLUSIONES .....	90
RECOMENDACIONES.....	91
BIBLIOGRAFÍA.....	92

## INTRODUCCIÓN

El sistema de péndulo invertido es un problema clásico en ingeniería de control que consiste en una varilla sujeta en uno de sus extremos a un eje que le permite pivotar en el plano, la cual va montada sobre un carro que se desplaza longitudinalmente. Se trata de un sistema inestable, ya que la varilla puede caer en cualquier momento hacia cualquier dirección dentro del plano. Para evitar que la varilla se caiga se introduce un controlador cuya misión es mantenerla en posición vertical hacia arriba. Para ello se debe tener control sobre la fuerza aplicada al carro, realimentando los estados necesarios en dependencia del controlador a utilizar.

En el Laboratorio de Automatización de la Universidad Nacional de Ingeniería UNI hacen falta herramientas donde se realicen prácticas de modelado, identificación de sistemas e implementación de la teoría de control abarcada en las asignaturas de Sistemas de Control y Control Aplicado lo que provoca que los estudiantes no cuenten con suficientes prácticas para fortalecer sus conocimientos sobre estos temas.

Por este motivo se propone como solución el diseño de un péndulo invertido con el objetivo de dotar al laboratorio con una plataforma física donde los estudiantes consoliden, enriquezcan, implementen y se motiven con la teoría de sistemas de control realimentados basados en modelos matemáticos.

Para llevar a cabo el diseño, el trabajo se estructuró en 2 capítulos. En el capítulo 1 “Marco teórico” se abordó conceptos generales referentes a sistemas, modelado, estimación de parámetros y algoritmos de control. En el capítulo 2 “Análisis y presentación de resultados” se hizo un análisis y se corroboró la necesidad que vendría a apaciguar el proyecto en el Laboratorio de Automatización. Se desarrolló la construcción del prototipo. Se modelaron y estimaron los parámetros del péndulo invertido y sus subsistemas. Se validó que los modelos capturarán la dinámica del sistema físico real y en base a estos modelos, se diseñaron e implementaron los controladores PID y LQR.



## JUSTIFICACIÓN

El péndulo invertido es un caso concreto de sistema físico en el que se ponen en manifiesto importantes problemas, lo que ha hecho de él un banco de pruebas para Sistemas de Control. El interés en el estudio de este sistema radica en que su modelo matemático presenta una formulación basada en ecuaciones diferenciales, que guarda cierta analogía con procesos reales de mayor complejidad tales como el modelo de control de posición en un impulsor espacial en el momento de despegue. El objeto del control de posición es mantener el impulsor espacial en posición vertical. El impulsor espacial real (o el péndulo invertido en este problema) es inestable y puede volcarse en cualquier momento y en cualquier dirección.

En este trabajo se propone el diseño, construcción y funcionalidad de un péndulo invertido con el fin proveer a la Universidad Nacional de Ingeniería, a docentes y estudiantes una herramienta más para ser utilizada en el desarrollo del contenido de las asignaturas de Sistemas de Control y Control Aplicado.

Por medio de este instrumento los estudiantes podrán relacionar el contenido desarrollado en las asignaturas anteriormente mencionadas, tratándose esto de un sistema en el que se abordan temas como el modelado matemático de los sistemas de control, función de transferencia, espacio de estados, identificación de parámetros, diseño de compensadores PID y LQR. Además podrán hacer uso de entornos de software para analizar la respuesta temporal del sistema a través de simulaciones.

Tanto estudiantes como docentes tendrán a su disposición un manual de usuario que explica el funcionamiento y manipulación del péndulo invertido, y que en base a este puedan realizar prácticas o experimentos para el estudio de los temas relacionados.

## **OBJETIVOS**

### **OBJETIVO GENERAL**

Diseñar un péndulo invertido para su uso en las asignaturas de Sistemas de Control y Control Aplicado de la Universidad Nacional de Ingeniería (UNI).

### **OBJETIVOS ESPECÍFICOS**

1. Construir un prototipo basado en el modelo del péndulo invertido montado en un carro.
2. Elaborar un modelo matemático del péndulo invertido en las representaciones de ecuaciones diferenciales, función de transferencia y espacio de estados, identificando los parámetros de las ecuaciones diferenciales mediante los métodos black-box y grey-box modeling.
3. Diseñar un compensador PID para el péndulo invertido, utilizando la técnica del lugar geométrico de las raíces, que permita que dicho péndulo mantenga una posición vertical simulando su respuesta temporal con Scilab, Octave o Matlab.
4. Diseñar un compensador LQR para el péndulo invertido, utilizando la técnica del espacio de estados, que permita que dicho péndulo mantenga una posición vertical simulando su respuesta temporal con Scilab, Octave o Matlab.
5. Implementar los compensadores diseñados en el péndulo invertido con un microcontrolador, encoders de cuadratura y motores, comparando el desempeño de ambos controladores.
6. Elaborar un manual de usuario del péndulo invertido para el docente de las asignaturas de Sistemas de Control y Control Aplicado de la Universidad Nacional de Ingeniería.

## CAPÍTULO 1: MARCO TEÓRICO

### 1.1 SISTEMA

Según la RAE un sistema es un “conjunto de reglas o principios sobre una materia racionalmente enlazados entre sí”.

Se puede representar un sistema mediante el diagrama de la figura 1, en el que nos permite observar cuáles son las entradas, qué salidas es necesario considerar y cómo estas salidas están relacionadas con las entradas. La forma de representar las relaciones entradas-salidas se conoce como modelo del sistema.



Figura 1. Sistema.

Fuente: Teoría de Sistemas y Circuitos, Lago & Benningfield, 1984.

El modelo del sistema es una abstracción. ¿Qué es abstraer? Se tiene por definición de la RAE que abstraer es “separar por medio de una operación intelectual un rasgo o una cualidad de algo para analizarlos aisladamente o considerarlos en su pura esencia o noción”.

En nuestro caso se estudian sistemas lineales de parámetros concentrados invariantes en el tiempo, en los que se utilizan las matemáticas para abstraer el sistema y crear un modelo matemático, también llamado modelo de caja blanca, representado por ecuaciones diferenciales ordinarias lineales con coeficientes constantes de la forma

$$a_n \frac{d^n x}{dt^n} + a_{n-1} \frac{d^{n-1} x}{dt^{n-1}} + \dots + a_1 \frac{dx}{dt} + a_0 x = f(t) \quad (1)$$

en donde el tiempo  $t$  es la variable independiente y  $x(t)$ <sup>1</sup> es la variable dependiente. Si  $f(t) = 0$  se dice que la ecuación es homogénea, mientras que si  $f(t) \neq 0$  entonces se dice que la ecuación es no homogénea. Se dice que la ecuación es de orden  $n$  porque  $n$  es el orden de la derivada más alta contenida en la ecuación. Se dice que esta ecuación es lineal porque tanto la variable dependiente  $x(t)$  como todas sus derivadas están elevadas a la primera potencia. Se dice que esta ecuación es de coeficientes constantes porque todos los factores  $a_n$  son constantes.

Existe una relación directa entre el término sistema lineal invariante en el tiempo de parámetros concentrados y el término ecuación diferencial ordinaria lineal con coeficientes constantes. La linealidad se ha descrito en el párrafo anterior. La invariancia en el tiempo se da por los coeficientes constantes. Lo de parámetros concentrados es por el término ordinaria, que significa que hay una sola variable independiente  $t$  y la variable dependiente  $x(t)$  varía solamente respecto al tiempo y no respecto al espacio dimensional (Lago & Benningfield, 1984).

Es importante mencionar que estos modelos son una aproximación de la realidad o del sistema bajo estudio, ya que la naturaleza se comporta de manera no lineal. Entonces, ¿por qué utilizar ecuaciones de la forma (1)? Bueno, por conveniencia, dada la facilidad de su análisis. Un importante principio de modelado compartido por el Dr. Ing. Burkhard Corves en el MOOC Machine Dynamics with Matlab en la plataforma Edx, es que un modelo tiene que ser tan simple como sea posible, pero tan complejo como sea necesario.

## 1.2 SOLUCIÓN COMPLEMENTARIA DE EDO<sup>2</sup> LINEALES

Dada la ecuación homogénea de la forma

$$a_n \frac{d^n x}{dt^n} + a_{n-1} \frac{d^{n-1} x}{dt^{n-1}} + \dots + a_1 \frac{dx}{dt} + a_0 x = 0 \quad (2)$$

---

<sup>1</sup> Se utiliza  $x$  o  $x(t)$  indistintamente por simplicidad.

<sup>2</sup> EDO: Ecuación Diferencial Ordinaria.

que se obtiene al hacer  $f(t) = 0$  en la ecuación (1), se le llama solución complementaria o respuesta libre del sistema, a la solución de la ecuación homogénea (2).

A diferencia de las ecuaciones algebraicas, la solución de las ecuaciones diferenciales son funciones en lugar de números. La solución  $x(t)$  en la ecuación (2) es una función elemental de la forma

$$x(t) = Ke^{st} \quad (3)$$

donde  $K$  y  $s$  son constantes. Para que  $x(t)$  sea una solución, es necesario que satisfaga la ecuación (2).

Al tomar la ecuación (3) y sus derivadas

$$\begin{aligned} x &= Ke^{st} \\ \frac{dx}{dt} &= Kse^{st} \\ \frac{d^2x}{dt^2} &= Ks^2e^{st} \\ \frac{d^nx}{dt^n} &= Ks^ne^{st} \end{aligned}$$

y sustituir estas ecuaciones en la ecuación (2), se observa que  $Ke^{st}$  es común a todos los términos y por lo tanto puede factorizarse como

$$(a_ns^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0)Ke^{st} = 0$$

El factor  $Ke^{st}$  no puede ser igual a cero, ya que representa a  $x(t)$  que es la solución buscada. Por lo tanto, para que la igualdad a cero se cumpla, es necesario que el término entre paréntesis sea igual a cero

$$(a_ns^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0) = 0 \quad (4)$$

A la ecuación (4) se le denomina ecuación característica o polinomio característico. Al tratar con polinomios es necesario acudir al teorema fundamental del álgebra el cual establece que: “cualquier polinomio de grado  $n$  con coeficientes complejos tiene exactamente  $n$  raíces, si cada raíz múltiple se cuenta tal número de veces, cual es su multiplicidad” (Tsipkin, 1985). Es decir, un polinomio de grado  $n$  tiene  $n$  cantidades de raíces, las cuales pertenecen al conjunto de los números complejos.

Al factorizar la ecuación característica (4) como se muestra a continuación

$$a_n(s + \sigma_1)(s + \sigma_2)(s + \sigma_3) \dots (s + \sigma_n) = 0 \quad (5)$$

se observa que hay  $i$  cantidades de raíces  $s_i = -\sigma_i$  igual al orden de la ecuación diferencial ordinaria lineal.

La solución general de la ecuación diferencial homogénea es igual a la suma de  $n$  funciones indicadas a continuación (Lago & Benningfield, 1984, págs. 19-21)

$$x(t) = K_1 e^{-\sigma_1 t} + K_2 e^{-\sigma_2 t} + \dots + K_n e^{-\sigma_n t} \quad (6)$$

o bien

$$x(t) = K_1 e^{s_1 t} + K_2 e^{s_2 t} + \dots + K_n e^{s_n t} \quad (7)$$

En resumen, las soluciones a las ecuaciones diferenciales ordinarias lineales con coeficientes constantes son exponenciales reales o complejas, transformándose estas últimas en sinusoides mediante la identidad de Euler  $e^{-j\theta} = \cos \theta + j \sin \theta$ . Esto hace que solamente sean necesarias las funciones elementales: exponenciales, senos y cosenos. Hay que recordar, que el seno es un coseno desfasado y viceversa (Lynch & Truxal, 1961).

### 1.3 SISTEMA GENERAL DE SEGUNDO ORDEN

Con el objeto de normalizar el estudio de los sistemas de segundo orden, o aquellos en cuyas respuestas libres están caracterizadas por una ecuación diferencial homogénea de segundo orden como la siguiente

$$\ddot{x} + a_1 \dot{x} + a_0 x = 0 \quad (8)$$

suelen designarse los coeficientes de la primera derivada  $\dot{x}$  y de la respuesta  $x$  con:

$$\alpha = \frac{1}{2} a_1 \quad (9)$$

$$\omega_0^2 = a_0$$

La constante  $\alpha$  recibe el nombre de constante de amortiguamiento o factor de amortiguamiento real; y  $\omega_0$ , de velocidad angular de oscilación no

amortiguada o de frecuencia natural. En otra literatura utilizan  $\omega_n$  para representar la frecuencia natural (Gerez Greisser & Murray-Lasso, 1972, págs. 222-223).

Sustituyendo las ecuaciones (9) en la (8) obtenemos

$$\ddot{x} + 2\alpha\dot{x} + \omega_0^2 x = 0 \quad (10)$$

Ésta es una ecuación diferencial homogénea de coeficientes constantes y su ecuación característica es

$$s^2 + 2\alpha s + \omega_0^2 = 0 \quad (11)$$

cuyas raíces son

$$\begin{aligned} s_1 &= -\alpha + \sqrt{\alpha^2 - \omega_0^2} \\ s_2 &= -\alpha - \sqrt{\alpha^2 - \omega_0^2} \end{aligned} \quad (12)$$

y reciben el nombre de frecuencias naturales del sistema.

La solución de la ecuación (11) depende de los valores de  $\alpha$  y  $\omega_0$  y tiene la forma de la ecuación (7)

$$x(t) = K_1 e^{s_1 t} + K_2 e^{s_2 t} \quad (13)$$

Las constantes  $K_1$  y  $K_2$  se determinan mediante las condiciones iniciales  $x(0)$  y  $\dot{x}(0)$  de la siguiente manera. Siendo

$$x(0) = K_1 e^{-s_1 \cdot 0} + K_2 e^{-s_2 \cdot 0} = K_1 + K_2 \quad (14)$$

y

$$\dot{x}(0) = K_1 s_1 e^{-s_1 \cdot 0} + K_2 s_2 e^{-s_2 \cdot 0} = K_1 s_1 + K_2 s_2 \quad (15)$$

resolvemos este sistema de ecuaciones multiplicando la ecuación (14) por  $-s_2$  y el resultado lo sumamos a la ecuación (15)

$$\begin{array}{rcl} -x(0)s_2 & = & -K_1 s_2 - K_2 s_2 \\ \dot{x}(0) & = & K_1 s_1 + K_2 s_2 \\ \hline \dot{x}(0) - s_2 x(0) & = & K_1 (s_1 - s_2) \end{array}$$

así obtenemos

$$K_1 = \frac{\dot{x}(0) - s_2 x(0)}{s_1 - s_2} \quad (16)$$

De la misma manera para  $K_2$  multiplicamos la ecuación (14) por  $-s_1$  y el resultado se lo sumamos a la ecuación (15)

$$\begin{array}{rcl} -s_1 x(0) & = & -K_1 s_1 - K_2 s_1 \\ \dot{x}(0) & = & K_1 s_1 + K_2 s_2 \\ \hline \dot{x}(0) - s_1 x(0) & = & K_2 (s_2 - s_1) \end{array}$$

así obtenemos

$$K_2 = \frac{\dot{x}(0) - s_1 x(0)}{s_2 - s_1} \quad (17)$$

#### 1.4 SISTEMA DE SEGUNDO ORDEN SUBAMORTIGUADO

Continuando con el sistema de segundo orden de la ecuación (10), en el caso subamortiguado se cumple la siguiente condición  $0 < \alpha < \omega_0$ .

Si tomamos las raíces de la ecuación (12), observamos que la cantidad debajo del radical es negativa, por lo que da apertura al número complejo  $j$ , quedando

$$\begin{aligned} s_1 &= -\alpha + j\sqrt{\omega_0^2 - \alpha^2} \\ s_2 &= -\alpha - j\sqrt{\omega_0^2 - \alpha^2} \end{aligned} \quad (18)$$

Si definimos

$$\omega_d = \sqrt{\omega_0^2 - \alpha^2} \quad (19)$$

las raíces de la ecuación (18) quedan

$$\begin{aligned} s_1 &= -\alpha + j\omega_d \\ s_2 &= -\alpha - j\omega_d \end{aligned} \quad (20)$$

las cuales son complejas conjugadas. El término  $\omega_d$  recibe el nombre de frecuencia natural amortiguada de oscilación o de velocidad angular amortiguada de oscilación y en algunos libros aparece solamente como  $\omega$ .

Al sustituir las raíces de la ecuación (20) en la (7) tenemos

$$x(t) = K_1 e^{(-\alpha + j\omega_d)t} + K_2 e^{(-\alpha - j\omega_d)t} \quad (21)$$

trabajando un poco la ecuación (21)

$$x(t) = e^{-\alpha t} (K_1 e^{j\omega_d t} + K_2 e^{-j\omega_d t}) \quad (22)$$

utilizando la identidad de Euler

$$e^{j\omega_d t} = \cos \omega_d t + j \sin \omega_d t$$



$$e^{-j\omega_d t} = \cos \omega_d t - j \sin \omega_d t$$

sustituyendo las identidades en la ecuación (22) y agrupando términos semejantes

$$x(t) = e^{-\alpha t} (K_1 \cos \omega_d t + K_1 j \sin \omega_d t + K_2 \cos \omega_d t - K_2 j \sin \omega_d t)$$

$$x(t) = e^{-\alpha t} [(K_1 + K_2) \cos \omega_d t + (K_1 - K_2) j \sin \omega_d t]$$

Dado que  $x(t)$  es real,  $K_1$  y  $K_2$  tienen que ser complejas conjugadas  $K_1 = \overline{K_2}$ .

Si definimos

$$\begin{aligned} A &= (K_1 + K_2) \\ B &= (K_1 - K_2)j \end{aligned} \quad (23)$$

obtenemos la solución del sistema de segundo orden subamortiguado

$$x(t) = e^{-\alpha t} [A \cos(\omega_d t) + B \sin(\omega_d t)] \quad (24)$$

La ecuación (24) se puede representar de otra forma. Si definimos

$$K_1 = c + jd$$

$A$  y  $B$  en la ecuación (23) quedarían

$$\begin{aligned} A &= K_1 + K_2 = K_1 + \overline{K_1} = 2c \\ B &= (K_1 - K_2)j = (K_1 - \overline{K_1})j = -2d \end{aligned} \quad (25)$$

sustituimos las ecuaciones (25) en la (24)

$$x(t) = 2e^{-\alpha t} [c \cdot \cos(\omega_d t) - d \cdot \sin(\omega_d t)] \quad (26)$$

Si representamos  $K_1$  mediante el siguiente triángulo

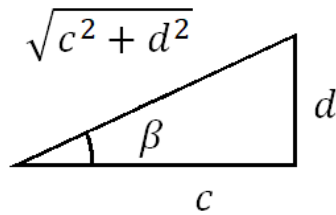


Figura 2. Definición del ángulo  $\beta$ .

Fuente: Teoría de Sistemas y Circuitos, Gerez Greisser & Murray-Lasso, 1972.

obtenemos la otra forma de representar la ecuación (24)

$$x(t) = 2 \sqrt{c^2 + d^2} e^{-\alpha t} [\cos(\omega_d t + \beta)] \quad (27)$$

Y

$$\beta = \tan^{-1} \left( \frac{d}{c} \right) \quad (28)$$

La ecuación (27) nos dice que la respuesta libre de un sistema de segundo orden subamortiguado es una senoide que decae en el tiempo, o dicho en otras palabras, son oscilaciones amortiguadas y se muestra en la figura 3 b.

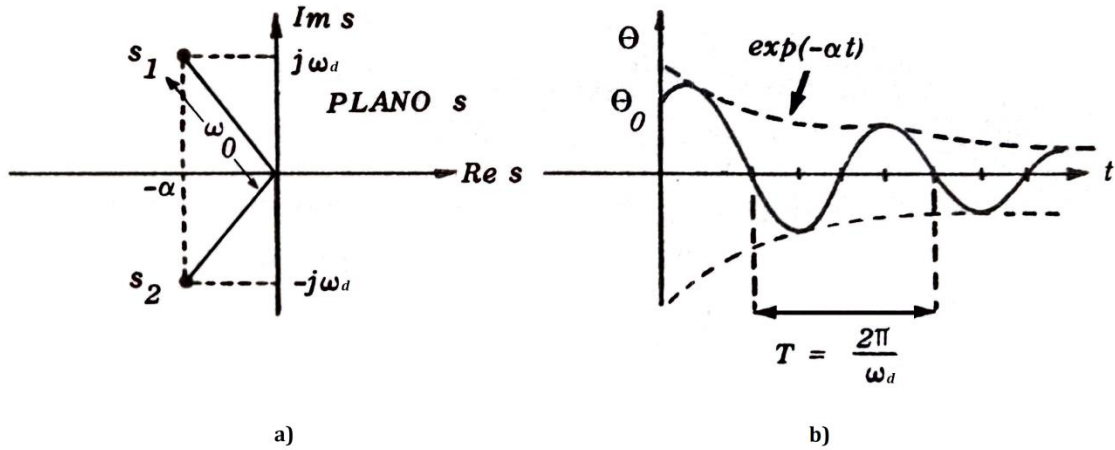


Figura 3. Sistema de segundo orden subamortiguado. a) Diagrama de las raíces  
b) Respuesta libre.

Fuente: Teoría de Sistemas y Circuitos, Gerez Greisser & Murray-Lasso, 1972.

Dado que las raíces son complejas, se utiliza el plano complejo  $s$  para graficarlas según se muestra en la figura 3 a. Es importante mencionar que  $\alpha$  nos dice cuán rápido decae y  $\omega_d$  nos dice cuán rápido oscila el sistema, donde  $T$  es el período de la oscilación amortiguada y se define mediante la siguiente ecuación (Gerez Greisser & Murray-Lasso, 1972, págs. 233-238)

$$T = \frac{2\pi}{\omega_d} \quad (29)$$

## 1.5 IDENTIFICACIÓN DE SISTEMAS

La identificación de sistemas consiste en acumular un número suficiente de observaciones sobre las señales de entrada y salida del sistema, con el propósito de emplearlas para construir un modelo del mismo. No se toma en cuenta el conocimiento interno del sistema, es decir, su representación en

ecuaciones diferenciales mediante las leyes de la física. Por lo tanto, la identificación del modelo del sistema se basa completamente en datos experimentales recopilados de la planta o proceso. El modelo obtenido mediante la identificación de sistemas se conoce como modelo de caja negra, modelo entrada – salida o black box models (Duarte V., pág. 4).

### 1.5.1 Grey-box Modelling

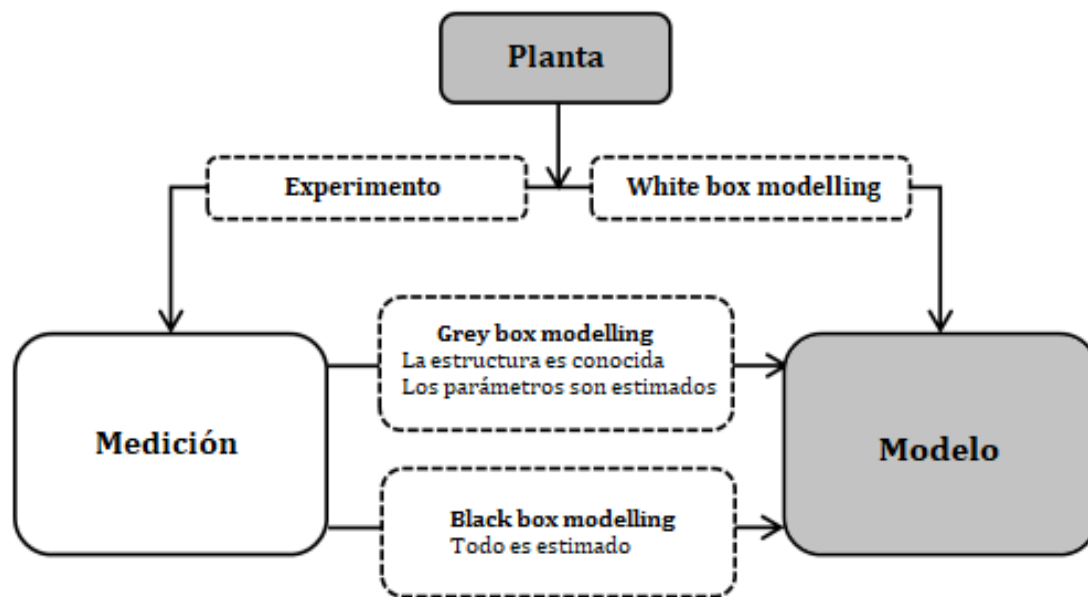


Figura 4. Construcción de modelos.

Fuente: Modelling and Simulation using Matlab, Patrick Metzler.

Grey-box modelling o modelado de caja gris, es un proceso híbrido de modelado de sistemas en el cual se emplea el conocimiento que esté a la mano acerca de la estructura interna del sistema y las leyes que rigen su comportamiento, y se emplean observaciones para determinar la información que haga falta. El modelo así obtenido se conoce como modelo de caja gris (Duarte V., pág. 4). En la figura 4, se puede observar, los tres tipos de modelado de sistemas. En el modelado de caja blanca o white box modelling, se conoce por completo la estructura interna del sistema a través de leyes físicas y ecuaciones diferenciales; en el modelado de caja negra se realiza una

identificación del sistema mediante señales de entrada-salida sin conocer la estructura interna del sistema; y en el modelado de caja gris, se utilizan ambos métodos para obtener un modelo del sistema. Es decir, si se tiene un modelo matemático en forma de ecuaciones diferenciales o ya se decidió la estructura de estas ecuaciones, se recopilan datos experimentales de la planta para estimar los parámetros o coeficientes de las ecuaciones.

## 1.6 OBTENCIÓN DE UN MODELO A BASE DE DATOS EXPERIMENTALES SEGUN EL MÉTODO DE CUADRADOS MÍNIMOS

Suponemos una estructura de un modelo de segundo orden como la siguiente que tiene forma de la ecuación (1)

$$p_2\ddot{y} + p_1\dot{y} + p_0y = u \quad (30)$$

es necesario estimar el vector  $p$  que contenga los coeficientes de la ecuación diferencial, que también son los coeficientes del polinomio característico.

Teniendo el siguiente arreglo matricial  $Ax = B$

$$\begin{bmatrix} \ddot{y}(t_1) & \dot{y}(t_1) & y(t_1) \\ \ddot{y}(t_2) & \dot{y}(t_2) & y(t_2) \\ \vdots & \vdots & \vdots \\ \ddot{y}(t_n) & \dot{y}(t_n) & y(t_n) \end{bmatrix} \cdot \begin{bmatrix} p_2 \\ p_1 \\ p_0 \end{bmatrix} = \begin{bmatrix} u(t_1) \\ u(t_2) \\ \vdots \\ u(t_n) \end{bmatrix} \quad (31)$$

la estimación del vector  $p$  se vuelve un problema de mínimos cuadrados para la matriz sobredeterminada de la ecuación (31). Se dice sobredeterminada porque hay  $n$  cantidades de ecuaciones lineales para  $m$  incógnitas, siendo  $n > m$ . En el caso de la ecuación (31)  $m = 3$ ; es decir,  $m$  es la cantidad de elementos del vector  $p$  que son los coeficientes de la ecuación diferencial.

La fórmula de los mínimos cuadrados es obtenida del problema

$$\min_x \|Ax - B\|$$

cuya solución se puede escribir

$$x = (A^T A)^{-1} A^T B$$

Dado que los datos son obtenidos experimentalmente, y muchas veces las derivadas no se pueden medir directamente, no es conveniente derivar para obtener las variables faltantes porque amplifica el ruido de las mediciones. Por

esta razón, en lugar de derivar se integra, teniendo cuidado solamente de no integrar sin límites, ya que de lo contrario se obtendrían constantes nefastas de integración que estarían en la matriz y no el vector  $p$ , de modo que no estaríamos hablando más de una ecuación lineal. Por consiguiente se debe integrar de cero a cierto tiempo  $t$ , partiendo de condiciones iniciales cero o estado estable (Metzler).

Así, el arreglo (31) queda

$$\begin{bmatrix} y(t_1) & \int y(t_1) & \iint y(t_1) \\ y(t_2) & \int y(t_2) & \iint y(t_2) \\ \vdots & \vdots & \vdots \\ y(t_n) & \int y(t_n) & \iint y(t_n) \end{bmatrix} \cdot \begin{bmatrix} p_2 \\ p_1 \\ p_0 \end{bmatrix} = \begin{bmatrix} \iint u(t_1) \\ \iint u(t_2) \\ \vdots \\ \iint u(t_n) \end{bmatrix} \quad (32)$$

## 1.7 SISTEMA DE CONTROL

Un sistema de control es simplemente cualquier sistema que controle un suministro de energía. En un sistema de control se tiene un sistema, planta o proceso, como el de la figura 1, en el cual se pretende que las salidas  $O_c$ , se comporten de una manera deseada, a través del control del suministro de energía al sistema en forma de las entradas  $I_R$ .

Por ejemplo, en un sistema de control de temperatura de una casa estando la temperatura en el exterior a 0 °C, se desea que la temperatura en el interior de la casa se comporte de una manera deseada a 24 °C, a través del control de suministro de energía térmica generada por un calefactor.

Específicamente, los sistemas de control lineales, con el objeto de preservar la linealidad, hacen uso solamente de las siguientes operaciones lineales: multiplicación por una constante, integración y derivación (Phelan, 1977).

## 1.8 SISTEMA DE CONTROL DE LAZO ABIERTO

En el sistema de control de lazo abierto, como se muestra en la figura 5, la salida del sistema no se compara con la entrada de referencia, por lo que es

necesaria una condición de operación fijada para que la salida se comporte de manera deseada. La exactitud del sistema depende de la calibración y ante perturbaciones el comportamiento no es el deseado (Ogata, 1980).

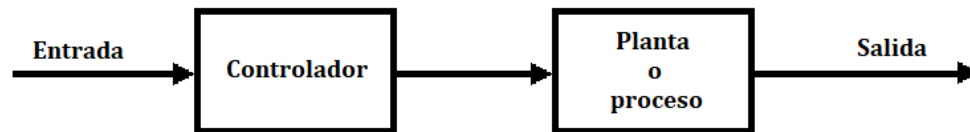


Figura 5. Sistema de control en lazo abierto.

Fuente: Ingeniería de Control Moderna, Ogata, 1980.

### 1.9 SISTEMA DE CONTROL DE LAZO CERRADO

En el sistema de control de lazo cerrado que se expone en la figura 6, la salida del sistema es realimentada, por lo que también son llamados sistemas de control realimentados. Esta salida realimentada tiene un efecto directo sobre la señal de control que se suministra a la planta o proceso. Se pretende con la realimentación de la salida, realizar una comparación con la entrada de referencia y reducir el error a cero entre ambas mediante un controlador. Así, la salida se asemejará a la entrada y se comportará acorde a lo deseado, mejorando su capacidad de superar las perturbaciones (Ogata, 1980).

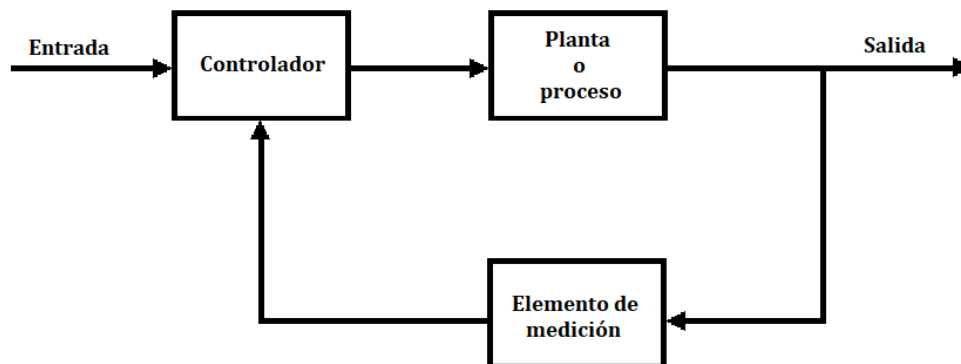


Figura 6. Sistema de control de lazo cerrado.

Fuente: Ingeniería de Control Moderna, Ogata, 1980.

### 1.10 CONTROLADOR PID

El control PID se puede expresar matemáticamente como

$$u(t) = K \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d(\tau) + T_d \frac{de(t)}{dt} \right) \quad (33)$$

La acción de control es así una suma de tres términos que representan el pasado por la acción integral del error (el término-I), el presente (el término-P) y el futuro por una extrapolación lineal del error (el término-D), como se muestra en la figura 7. El término  $e + T_d \frac{de}{dt}$  es una predicción lineal del error  $T_d$  unidades de tiempo en el futuro. Los parámetros del controlador se llaman: ganancia proporcional  $K$ , tiempo integral  $T_i$ , y tiempo derivativo  $T_d$  (J. Astrom & Hagglund, 2009).

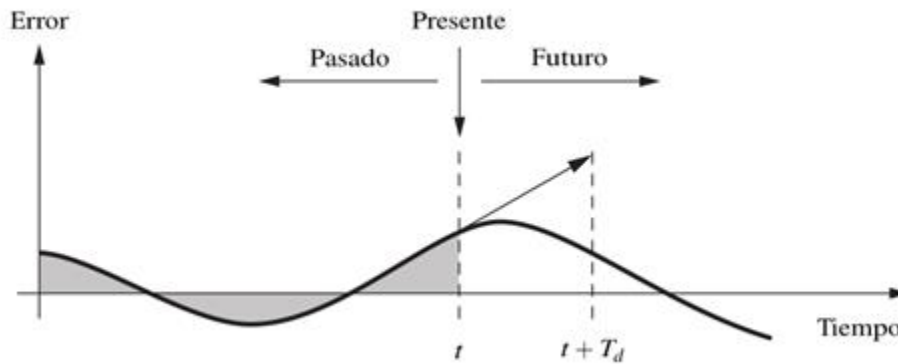


Figura 7. Controlador PID.

Fuente: Control PID Avanzado, J. Astrom & Hagglund, 2009.

### 1.11 CONTROLADOR LQR

Debe sus siglas al nombre en inglés Linear Quadratic Regulator (Regulador Lineal Cuadrático). En la teoría de control óptimo, un regulador cuadrático lineal LQR consiste en un control realimentado de la forma

$$u = Kx$$

que minimice el siguiente índice de comportamiento

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

donde  $Q$  es una matriz positiva semidefinida y  $R$  es una matriz positiva definida. La matriz  $Q$  representa el peso asociado con el estado, mientras  $R$  representa el peso asociado a la entrada.

Bajo estos supuestos es posible encontrar que:

$$K = -R^{-1}B^TP$$

donde  $P$  es la solución a la ecuación de Ricatti

$$A^TP + PA - PBR^{-1}B^TP + Q = 0$$

Para poder desarrollar este tipo de controlador es necesario verificar su controlabilidad. La controlabilidad significa que para cualquier  $t_1 > t_0$  y para cualquier estado  $x_0$  y  $x_1$  existe una entrada  $u(t)$  tal que el sistema puede ir desde su estado inicial  $x_0(t_0)$  al estado final  $x_1(t_1)$ . Esto permite que el sistema converja hacia un estado deseado dado que es posible ubicar los polos del sistema en cualquier parte del plano complejo (Venturin & Poles).

## 1.12 EL PÉNDULO INVERTIDO SOBRE EL CARRO

Para el estudio del péndulo invertido<sup>3</sup> empezamos especificando el sistema que se muestra en la figura 8, donde un péndulo invertido está atado a un pivote  $P$  montado sobre un carro impulsado por un motor. En este caso consideramos sólo el problema en dos dimensiones, en que el péndulo se mueve únicamente en el plano del papel. El péndulo invertido es inestable ya que se puede caer en cualquier momento a menos de que se aplique una fuerza de control adecuada. Suponemos que la masa del péndulo está concentrada en el extremo y que la varilla no tiene masa. La fuerza de control  $u$  se aplica al carro. Este sistema es de considerable interés porque representa un laboratorio útil de idealización de sistemas mecánicos inestables que son encontrados de vez en cuando y que deben ser controlados.

---

<sup>3</sup> También se utiliza “péndulo invertido” para referirse solamente a la varilla en su configuración hacia arriba, sin el carro.



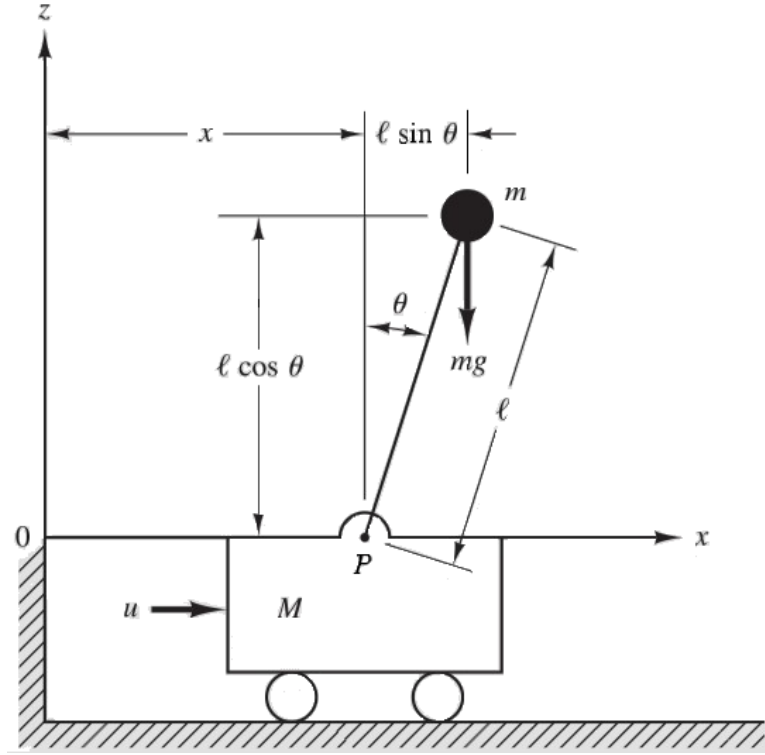


Figura 8. Sistema del péndulo invertido.

Fuente: Sistemas de Control en Tiempo Discreto, Ogata, 1996.

En el diagrama,  $\theta$  es el ángulo de la varilla respecto a la línea vertical. Suponemos que  $\theta$  es pequeño por lo que  $\sin \theta$  y el  $\cos \theta$  se pueden aproximar a  $\theta$  y a 1 respectivamente, y también suponemos que  $\dot{\theta}$  es pequeño por lo que  $\dot{\theta}^2 \approx 0$ . Estas aproximaciones se pueden explicar mediante la expansión de Taylor/Maclaurin de estas funciones. Tomaremos como ejemplo un ángulo  $\theta$  de 0.17 radianes (10 grados aproximadamente). Siendo los primeros tres términos de la expansión

$$\sin \theta = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \dots \quad (34)$$

$$\cos \theta = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \dots \quad (35)$$

Dado que  $\theta$  es pequeño, el segundo  $\left(\frac{\theta^3}{3!} \approx 0.0008\right)$  y tercer término  $\left(\frac{\theta^5}{5!} \approx 1.3 \times 10^{-6}\right)$  son mucho más pequeños que  $\theta$ . Por lo tanto, podemos observar que los términos de mayor orden se hacen aún más pequeños

aproximándose cada vez más a cero pudiéndose despreciar. Así quedan aproximadas las expansiones de  $\sin \theta \approx \theta$  y  $\cos \theta \approx 1$  respectivamente. Bajo estas condiciones, las ecuaciones del sistema no lineal se pueden linealizar.

### 1.12.1 Modelado del péndulo invertido

Se definen las coordenadas  $(x, y)$  del centro de gravedad de la masa como  $(x_G, z_g)$ . Entonces

$$x_G = x + l \sin \theta$$

$$z_G = l \cos \theta$$

Al aplicar la segunda ley de Newton en la dirección  $x$  del movimiento se obtiene

$$M \frac{d^2 x}{dt^2} + m \frac{d^2 x}{dt^2} = u$$

o

$$M \frac{d^2 x}{dt^2} + m \frac{d^2}{dt^2} (x + l \sin \theta) = u \quad (36)$$

Observamos que

$$\frac{d}{dt} \sin \theta = (\cos \theta) \dot{\theta}$$

$$\frac{d^2}{dt^2} \sin \theta = -(\sin \theta) \dot{\theta}^2 + (\cos \theta) \ddot{\theta}$$

$$\frac{d}{dt} \cos \theta = -(\sin \theta) \dot{\theta}$$

$$\frac{d^2}{dt^2} \cos \theta = -(\cos \theta) \dot{\theta}^2 - (\sin \theta) \ddot{\theta}$$

La ecuación (36) se puede escribir como

$$(M + m) \ddot{x} - ml(\sin \theta) \dot{\theta}^2 + ml(\cos \theta) \ddot{\theta} = u \quad (37)$$

La ecuación del movimiento de la masa  $m$  en la dirección  $z$  no se puede escribir sin considerar el movimiento de la masa  $m$  en la dirección  $x$ . Por lo que, en lugar de considerar el movimiento de la masa  $m$  en la dirección  $z$ , se considera el movimiento rotacional de la masa  $m$  alrededor del punto  $P$ . Al aplicar la segunda ley de Newton al movimiento rotacional, se obtiene

$$m \frac{d^2 x_G}{dt^2} l \cos \theta - m \frac{d^2 z_G}{dt^2} l \sin \theta = mgl \sin \theta$$

o

$$\left[ m \frac{d^2}{dt^2} (x + l \sin \theta) \right] l \cos \theta - \left[ m \frac{d^2}{dt^2} (l \cos \theta) \right] l \sin \theta = mgl \sin \theta$$

que se puede simplificar como sigue:

$$m[\ddot{x} - l(\sin \theta)\dot{\theta}^2 + l(\cos \theta)\ddot{\theta}]l \cos \theta - m[-l(\cos \theta)\dot{\theta}^2 - l(\sin \theta)\ddot{\theta}]l \sin \theta = mgl \sin \theta$$

Una simplificación posterior resulta en

$$m\ddot{x} \cos \theta + ml\ddot{\theta} = mg \sin \theta \quad (38)$$

Al sustituir el  $\sin \theta \approx \theta$ ,  $\cos \theta \approx 1$  y  $\theta \dot{\theta} \approx 0$ , las ecuaciones (37) y (38) se pueden linealizar como sigue:

$$(M + m)\ddot{x} + ml\ddot{\theta} = u \quad (39)$$

$$m\ddot{x} + ml\ddot{\theta} = mg\theta \quad (40)$$

Estas ecuaciones linealizadas son válidas mientras  $\theta$  y  $\dot{\theta}$  sean pequeñas. Las ecuaciones (39) y (40) definen un modelo matemático del sistema del péndulo invertido y tienen la forma de la ecuación (1) (Ogata, 1996, págs. 625-627).

### 1.13 EL PÉNDULO SOBRE EL CARRO

Considérese el péndulo sobre el carro que se muestra en la figura 9. Se asume que no hay fricción entre el carro y el marco de referencia y que el carro se mueve hacia los lados gracias a una fuerza externa  $u$ . Este es un sistema de dos grados de libertad. Las coordenadas generalizadas son  $x$  y  $\theta$ . La energía cinética del sistema es

$$T = \frac{1}{2} M v_1^2 + \frac{1}{2} m v_2^2$$

donde  $v_1 = \dot{x}$  y  $v_2$  es la velocidad absoluta de la masa  $m$ . El cuadrado de la velocidad  $v_2$  de la masa  $m$  se puede obtener mediante (Ogata, Dinámica de Sistemas, 1987)

$$v_2^2 = (\dot{x} + l \cos \theta \dot{\theta})^2 + (l \sin \theta \dot{\theta})^2$$

$$= \dot{x}^2 + l^2 \dot{\theta}^2 + 2\dot{x}l \cos \theta \dot{\theta}$$

Por lo tanto, la energía cinética es

$$T = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m(\dot{x}^2 + l^2\dot{\theta}^2 + 2\dot{x}l \cos \theta \dot{\theta})$$

La energía potencial del sistema es

$$U = mgl(1 - \cos \theta)$$

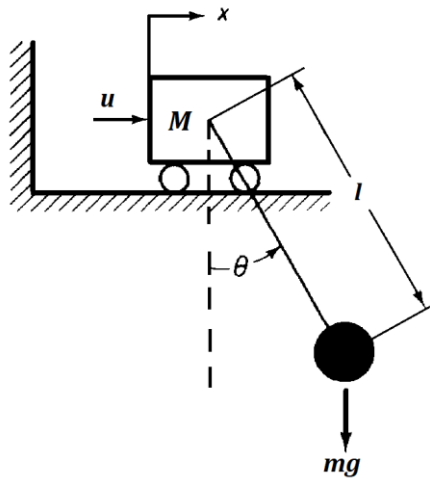


Figura 9. Péndulo sobre el carro.

Fuente: Dinámica de Sistemas, Ogata, 1987.

donde la energía potencial cuando  $x = 0$  y  $\theta = 0$  se toma como cero.

El Lagrangiano  $L$  es

$$L = T - U$$

$$L = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m(\dot{x}^2 + l^2\dot{\theta}^2 + 2\dot{x}l \cos \theta \dot{\theta}) - mgl(1 - \cos \theta)$$

Por lo tanto, las ecuaciones de Lagrange

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} = u$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = 0$$

se hacen

$$\frac{d}{dt}(M\dot{x} + m\dot{x} + ml \cos \theta \dot{\theta}) = u$$

$$\frac{d}{dt}(ml^2\ddot{\theta} + m\dot{x}l \cos \theta) + m\dot{x}l \sin \theta \dot{\theta} + mgl \sin \theta = 0$$

o bien

$$(M + m)\ddot{x} + ml(\cos \theta \ddot{\theta} - \sin \theta \dot{\theta}^2) = u \quad (41)$$

$$\ddot{\theta} + \frac{1}{l}\ddot{x} \cos \theta + \frac{g}{l} \sin \theta = 0 \quad (42)$$

#### 1.14 MODELADO MATEMÁTICO DEL CARRO

Una característica esencial del análisis de ingeniería de sistemas lineales es la simplificación que resulta de la técnica de dividir un problema complejo en varios problemas simples, y la capacidad resultante del ingeniero para lograr una comprensión de las propiedades y el comportamiento del sistema general. (Lynch & Truxal, 1961)

Por lo tanto, dada la complejidad del péndulo invertido, se sigue esta técnica y se divide el sistema general en subsistemas para su estudio de manera independiente. Estos subsistemas son: el carro y el péndulo.

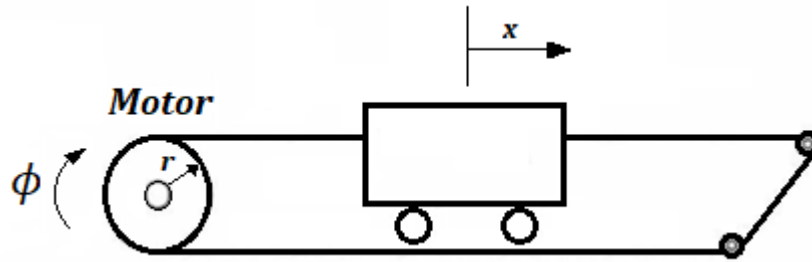


Figura 10. Carro impulsado por motor.

Fuente: Autor.

El carro está impulsado por un motor como se muestra en la figura 10. El movimiento rotacional del eje del motor se traduce en el movimiento traslacional del carro mediante una correa. La traducción de un tipo de movimiento a otro, no es más que la relación lineal dada por la fórmula de la longitud del arco

$$x = r\phi \quad (43)$$

donde  $x$  representa la longitud del arco o el desplazamiento traslacional del carro en este caso,  $\phi$  el ángulo en radianes o el desplazamiento angular del eje del motor y  $r$  la longitud del radio del eje del motor.

Por consiguiente, el estudio del carro es esencialmente el estudio del motor.

### 1.14.1 Modelado del motor

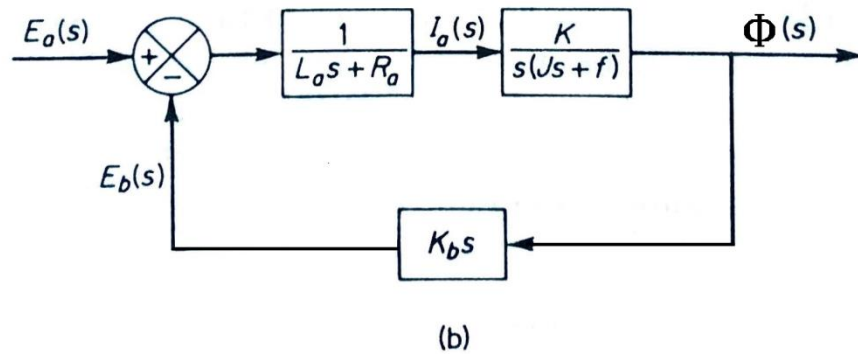
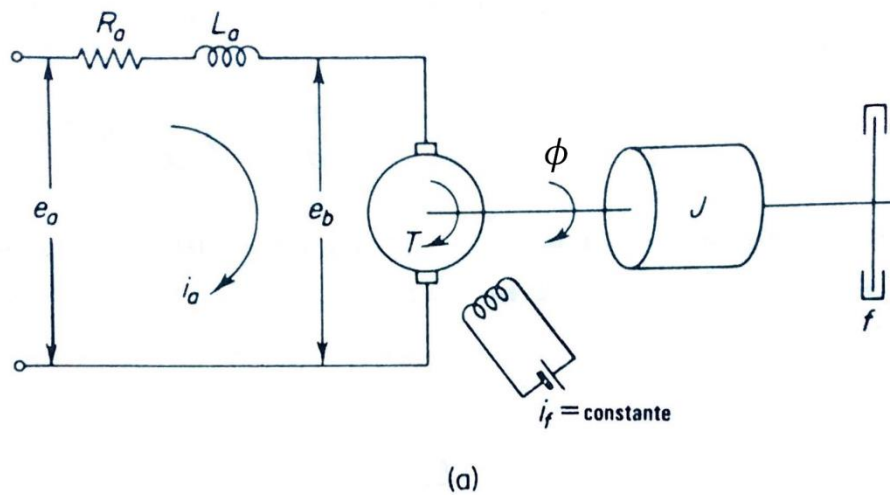


Figura 11. (a) Diagrama esquemático de un motor de cc controlado por el inducido. (b) Diagrama de bloques.

Fuente: Ingeniería de Control Moderna, Ogata, 1980.

Sea el motor de cc controlado en el inducido que puede verse en la figura 11 (a). En el sistema,

$R_a$  = resistencia del devanado del inducido en ohmios

$L_a$  = inductancia del devanado del inducido en henrios

$i_a$  = corriente del devanado del inducido en amperios

$i_f$  = corriente de campo en amperios

$e_a$  = tensión aplicada en la armadura en voltios

$e_b$  = fuerza contra-electromotriz en voltios

$\phi$  = desplazamiento angular del eje del motor en radianes

$T$  = par desarrollado por el motor en libras-pie

$J$  = momento de inercia equivalente del motor y carga con referencia al eje del motor en libras-pi/rad/seg

El par  $T$  desarrollado por el motor es proporcional al producto de la corriente del inducido  $i_a$  y el flujo  $\psi$  del entrehierro que a su vez es proporcional a la corriente del campo o

$$\psi = K_f i_f$$

donde  $K_f$  es una constante. Por tanto, se puede escribir el par  $T$

$$T = K_f i_f K_1 i_a$$

donde  $K_1$  es una constante.

En el motor de cc controlado en el inducido, se mantiene constante la corriente del campo. Para una corriente de campo constante el flujo se vuelve constante y el par se hace directamente proporcional a la corriente del inducido, de manera que

$$T = K i_a$$

donde  $K$  es una constante del par motor. Cuando el inducido está en rotación, se induce en el inducido una tensión proporcional al producto del flujo por la velocidad angular. Para un flujo constante, la tensión inducida  $e_b$  es directamente proporcional a la velocidad angular  $\frac{d\phi}{dt}$ . Así,

$$e_b = K_b \frac{d\phi}{dt} \quad (44)$$

donde  $K_b$  es una constante de fuerza contraelectromotriz.

Se controla la velocidad de un motor de cc controlado por el inducido por medio de la tensión del inducido  $e_a$ . La ecuación diferencial del circuito de inducido es

$$L_a \frac{di_a}{dt} + R_a i_a + e_b = e_a \quad (45)$$

La corriente del inducido produce el par que aplica a la inercia y fricción; por tanto

$$J \frac{d^2\phi}{dt^2} + f \frac{d\phi}{dt} = T = K i_a \quad (46)$$

Suponiendo que todas las condiciones iniciales son cero, y tomando las transformadas de Laplace de las ecuaciones (44), (45) y (46), se obtienen las siguientes ecuaciones:

$$K_b s \Phi(s) = E_b(s) \quad (47)$$

$$(L_a s + R_a) I_a(s) + E_b(s) = E_a(s) \quad (48)$$

$$(J s^2 + f s) \Phi(s) = T(s) = K I_a(s) \quad (49)$$

Considerando a  $E_a(s)$  como la entrada y  $\Phi(s)$  como la salida, se puede construir el diagrama de bloques de las ecuaciones (47), (48) y (49), como puede verse en la figura 11 (b). Se ve el efecto de la fuerza contraelectromotriz en la señal de realimentación proporcional a la velocidad del motor. Esta fuerza contraelectromotriz, por tanto, aumenta el amortiguamiento efectivo del sistema. Se obtiene la función transferencia de este sistema como

$$\frac{\Phi(s)}{E_a(s)} = \frac{K}{s[L_a J s^2 + (L_a f + R_a J)s + R_a f + K K_b]} \quad (50)$$

La inductancia  $L_a$  en el circuito de inducido generalmente es pequeña y se la puede despreciar. Si se desprecia  $L_a$ , la función de transferencia dada por la ecuación (50) se reduce a

$$\frac{\Phi(s)}{E_a(s)} = \frac{K_m}{s(T_m s + 1)} \quad (51)$$

donde



$$K_m = \frac{K}{(R_a f + K K_b)} = \text{constante de ganancia del motor}$$

$$T_m = \frac{R_a J}{(R_a f + K K_b)} = \text{constante de tiempo del motor}$$

De las ecuaciones (50) y (51) se puede ver que las funciones transferencias involucran el término  $\frac{1}{s}$ . Por lo tanto, este sistema posee una propiedad integrativa (Ogata, 1980, págs. 108-110).

La ecuación diferencial del motor en base a la función de transferencia de la ecuación (51) es

$$T_m \ddot{\phi} + \dot{\phi} = K_m e_a \quad (52)$$

donde  $\dot{\phi}$  y  $\ddot{\phi}$  son la velocidad y la aceleración angular respectivamente, o la primera y segunda derivada de la posición angular respecto al tiempo según la notación de Newton y tiene la forma de la ecuación (1).

### 1.15 MODELADO MATEMÁTICO DEL PÉNDULO

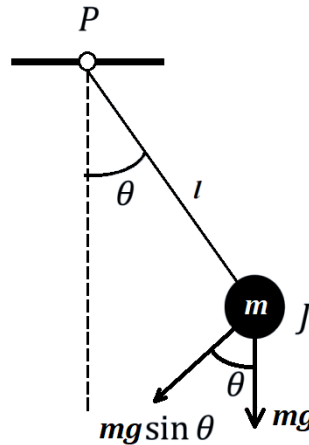


Figura 12. Péndulo.

Fuente: Autor.

El sistema del péndulo, el cual se muestra en la figura 12, consiste en una masa  $m$  que cuelga de un pivote  $P$  a través de una cuerda de longitud  $l$ . Se asume que la cuerda no tiene masa y que está recta en todo momento. También se asume que la vertical corresponde a un ángulo de cero grados y que hay fricción en el pivote. El péndulo oscila de lado a lado cuando se le suelta de un

ángulo distinto de cero, hasta regresar a su punto de equilibrio o cero grados. En el péndulo se da una transformación entre energías cinética y potencial, y parte de la energía se disipa en su forma térmica. Éste tipo de conversión no reversible de energía térmica es debido al fenómeno de fricción.

Para simplificar la forma del modelo del péndulo se supone que las acciones o efectos son de carácter concentrado, ya que esto permite utilizar parámetros concentrados, y por consiguiente, desarrollar el modelo en forma de ecuación diferencial ordinaria. Por lo tanto, en el modelo del péndulo se asume que la masa  $m$  está concentrada en un punto.

Para empezar el modelado, se supone que el péndulo está en reposo y se escoge el sentido anti-horario como dirección positiva de giro. Debido a que solo hay un punto de giro, que es el pivote  $P$ , es necesaria una sola ecuación diferencial para describir al sistema, también llamada ecuación de movimiento.

Se dibuja el siguiente diagrama de cuerpo libre

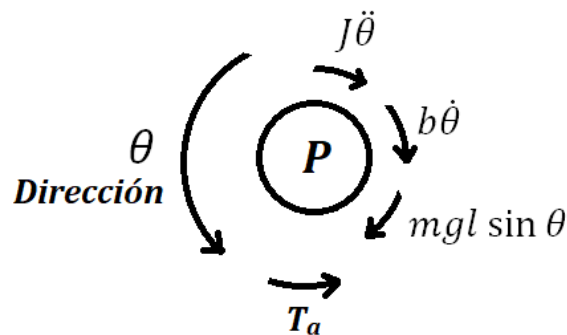


Figura 13. Diagrama de cuerpo libre del péndulo.

Fuente: Autor.

Suponemos que al péndulo le aplicamos un par positivo  $T_a$ , por lo cual éste gira positivamente. En oposición al movimiento actúa el par debido a la gravedad  $mgl \sin \theta$ , que se determina mediante la componente tangencial de la fuerza de gravedad  $mg \sin \theta$  por la longitud de la cuerda  $l$ ; el par debido a la fricción viscosa  $b\dot{\theta}$  y el par debido a la inercia  $J\ddot{\theta}$ .

Ahora escribimos la ecuación diferencial de movimiento usando la ley de Newton para sumar a cero todos los pares que se ilustran alrededor del punto de giro  $P$  en la figura 13 (Nise, 2009):

$$J\ddot{\theta} + b\dot{\theta} + mgl \sin \theta = T_a \quad (53)$$

Dado que la masa está concentrada en un solo punto, a una distancia  $l$  del punto de giro  $P$ , el momento de inercia para la rotación del eje es

$$J = ml^2 \quad (54)$$

Sustituyendo la ecuación (54) en la (53) se obtiene el modelo matemático o la ecuación no homogénea del péndulo

$$ml^2\ddot{\theta} + b\dot{\theta} + mgl \sin \theta = T_a \quad (55)$$

Haciendo  $T_a = 0$  y dividiendo entre  $ml^2$  cada uno de los términos de la ecuación (55), obtenemos la ecuación homogénea del péndulo (Cannon, 1967)

$$\ddot{\theta} + \frac{b}{ml^2}\dot{\theta} + \frac{g}{l} \sin \theta = 0 \quad (56)$$

## 1.16 MODELADO MATEMÁTICO DEL PÉNDULO INVERTIDO

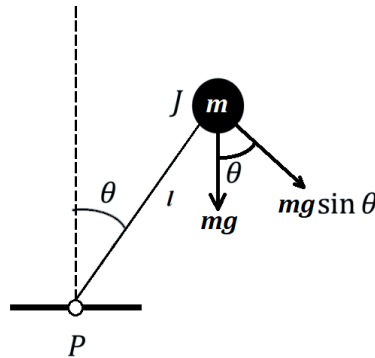


Figura 14. Péndulo invertido.

Fuente: Autor.

La transición del péndulo al péndulo invertido, no es más que el cambio de signo del torque debido a la gravedad.

Para el modelado se siguen los mismos pasos que en el caso del péndulo hacia abajo. Asumimos que el péndulo invertido está en reposo en su posición

vertical hacia arriba o cero grados, y se escoge el sentido horario como dirección positiva de giro.

Dibujamos el diagrama de cuerpo libre

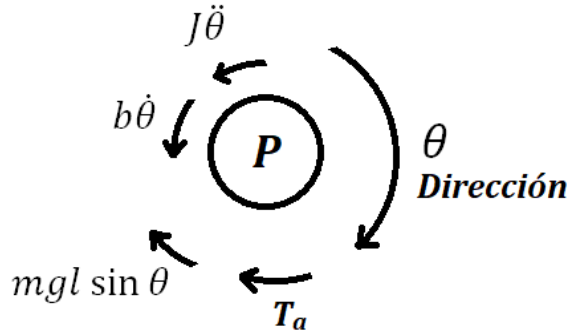


Figura 15. Diagrama de cuerpo libre del péndulo invertido.

Fuente: Autor.

Suponemos que le aplicamos al péndulo invertido un par positivo  $T_a$ , por lo cual éste gira positivamente. En el sistema actúan los mismos pares que en el caso del péndulo, por lo que de la misma manera, escribimos la ecuación diferencial de movimiento usando la ley de Newton para sumar a cero todos los pares alrededor del punto de giro  $P$  como se muestra en la figura 15 (Nise, 2009):

$$ml^2\ddot{\theta} + b\dot{\theta} - mgl \sin \theta = T_a \quad (57)$$

Haciendo  $T_a = 0$  y dividiendo entre  $ml^2$  cada uno de los términos de la ecuación (57), obtenemos el modelo matemático o la ecuación homogénea del péndulo invertido (Cannon, 1967)

$$\ddot{\theta} + \frac{b}{ml^2}\dot{\theta} - \frac{g}{l}\sin \theta = 0 \quad (58)$$

### 1.17 ESPACIO DE ESTADOS

El estado de un sistema dinámico es el conjunto más pequeño de variables (denominadas variables de estado) tales que el conocimiento de esas variables en  $t = t_0$ , juntamente con la entrada para  $t \geq t_0$ , determinan totalmente el comportamiento del sistema para cualquier tiempo  $t \geq t_0$ .

Para un sistema de orden  $n$ , se tiene un conjunto de  $n$  números que se representan como un vector columna y se le da el nombre de vector de estado del sistema, o bien, debido a que los valores de estos número cambian con el tiempo, a los componentes del vector de estado se les refiere como variables de estado. La notación utilizada para las variables de estado es  $x_j(t)$  en donde  $j = 1, 2, 3, \dots, n$ , y para el vector de estado

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}$$

Así, un conjunto de variables de estado es un número mínimo de variables del sistema cuyos valores para un tiempo determinado  $t_0$ , junto con el conocimiento de las entradas del sistema para  $t \geq t_0$ , determina de manera única los valores de estas mismas variables para  $t \geq t_0$ .

El espacio de estados es el espacio  $n$ -dimensional, cuyos ejes de coordenadas consisten en el eje  $x_1$ , eje  $x_2, \dots$ , eje  $x_n$ . Se puede representar cualquier estado por un punto en el espacio de estados.

El modelo dinámico de un sistema lineal en términos de la ecuación diferencial de vectores de estado es

$$\dot{x} = Ax + Bu \quad (59)$$

La representación de un sistema en variables de estado, no es más que otra manera de representar las ecuaciones diferenciales del modelo matemático del sistema. Si se tiene una ecuación diferencial de orden  $n$ , entonces, se representa está ecuación en notación matricial en la forma de la ecuación (59), en términos de  $n$  ecuaciones diferenciales de primer orden.

Esto significa, que al tener un sistema representado por la ecuación diferencial de enésimo orden siguiente:

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_{n-1} \frac{dy}{dt} + a_n y = u \quad (60)$$

donde la función excitadora  $u$  no involucra términos derivativos, se puede tomar a  $y, \frac{dy}{dt}, \dots, \frac{d^{n-1} y}{dt^{n-1}}$  como un conjunto de  $n$  variables de estado y se define (Ogata, Ingeniería de Control Moderna, 1980)

$$\begin{aligned}
x_1 &= y \\
x_2 &= \frac{dy}{dt} \\
&\vdots \\
x_n &= \frac{d^{n-1}y}{dt^{n-1}}
\end{aligned}$$

Entonces se puede escribir la ecuación (60) como

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
&\vdots \\
\dot{x}_{n-1} &= x_n \\
\dot{x}_n &= -a_n x_1 - a_{n-1} x_2 - \dots - a_1 x_n + u
\end{aligned}$$

o bien en la forma de la ecuación (59)

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}}_x + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}}_B u \quad (61)$$

siendo la ecuación de salida

$$y = Cx = [1 \ 0 \ \dots \ 0] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} \quad (62)$$

Un sistema es completamente controlable, si y solo si, el rango de la matriz  $n \times n$  (63) es  $n$ .

$$[B|AB|\dots|A^{n-1}B] \quad (63)$$

Ahora, un sistema es completamente observable, si y solo si, el rango de la matriz  $nm \times n$  (64) es  $n$ .

$$\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (64)$$

### 1.18 SCILAB

Scilab es una herramienta de software libre, de código abierto orientado al cálculo numérico; de sintaxis sencilla para el manejo de matrices y todo tipo de

operaciones con ellas, números complejos, polinomios y muchas otras funciones matemáticas.

Scilab es un excelente programa utilizado especialmente para aplicaciones científicas y de ingeniería, de fácil acceso pues su licencia GPL, permite su libre distribución, modificación y uso de software dando además de ello la disponibilidad de su última versión vía Internet. (Calvo Rolle, 2010)

### **1.19 MATLAB**

MATLAB combina un entorno de escritorio perfeccionado para el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices y arrays directamente.

Las capacidades de MATLAB se amplían mediante toolboxes que se desarrollan de forma profesional, pasan pruebas rigurosas y están totalmente documentadas. Además de apps que permiten ver cómo funcionan diferentes algoritmos y sus datos, en los cuales se pueden realizar iteraciones hasta obtener los resultados deseados y, después, generar automáticamente un programa para reproducir o automatizar el trabajo; es decir, la conversión automática de algoritmos en código C/C++ y HDL para su ejecución en dispositivos embebidos.

Una importante característica de MATLAB es que colabora con Simulink para permitir el diseño basado en modelos, el cual se emplea para la simulación multidominio (Matlab. Descripción general).

### **1.20 SIMULINK**

Es un entorno de diagramas de bloque para la simulación multidominio y el diseño basado en modelos. Admite el diseño y la simulación a nivel de sistema, la generación automática de código y la prueba y verificación continuas de los sistemas embebidos.

Simulink ofrece un editor gráfico, bibliotecas de bloques personalizables y solvers para modelar y simular sistemas dinámicos. Se integra con MATLAB<sup>®</sup>, lo que permite incorporar algoritmos de MATLAB en los modelos y exportar los

resultados de la simulación a MATLAB para llevar a cabo más análisis (Simulink. Descripción general).

## **1.21 ARDUINO**

Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Está compuesta de dos partes principales: la tarjeta Arduino, como en la figura 16, que es la pieza de hardware que se utiliza en la construcción del prototipo; y el Arduino IDE, que es la pieza de software que corre en la computadora, donde se crea el programa o sketch que le dice a la tarjeta qué hacer.

El hardware o la tarjeta Arduino, es una pequeña tarjeta microcontroladora o circuito que contiene un microcontrolador y todos los componentes necesarios para que éste trabaje correctamente y se pueda comunicar con la computadora.

El software o IDE, debe sus siglas del inglés Integrated Development Enviroment, o Entorno de Desarrollo Integrado. Es un programa que corre en la pc y permite escribir el programa o sketch que se carga en la tarjeta Arduino para que ésta lo ejecute. El código escrito en el sketch es traducido a lenguaje C, y pasado al compilador avr-gcc para ser finalmente traducido a lenguaje máquina del microcontrolador. Para la programación del Arduino existen muchas alternativas, entre ellas están Scilab y Matlab, siendo el último utilizado en este proyecto.

El Arduino Mega 2560 es una de las variadas tarjetas que tiene Arduino y se muestra en la figura 16. Tiene una mayor densidad de puertos, mayor memoria y mejores características que la tarjeta más conocida, Arduino UNO. Se basa específicamente en el microcontrolador ATMEGA2560-16AU (Banzi, 2011).





Fuente: Fritzing.

Los codificadores incrementales o incremental encoders, convierten el desplazamiento lineal o rotatorio en señales de pulsos. Proveen un pulso por cada incremento de resolución pero no hace distinciones entre dichos incrementos. Un codificador incremental rotatorio se presenta en la figura 17.



Fuente: Sistemas de Control Automático, Kuo, 1996.

La forma de onda de salida de los codificadores incrementales compatibles con la lógica digital se muestra en la figura 17. Específicamente esta señal cuadrada es de un codificador de dos canales,  $A$  y  $B$ , lo que permite además detectar direcciones. Cuando estos dos trenes de pulsos tienen un

desfase de  $90^\circ$  uno del otro, se dice que las dos señales están en cuadratura (Kuo, 1996).

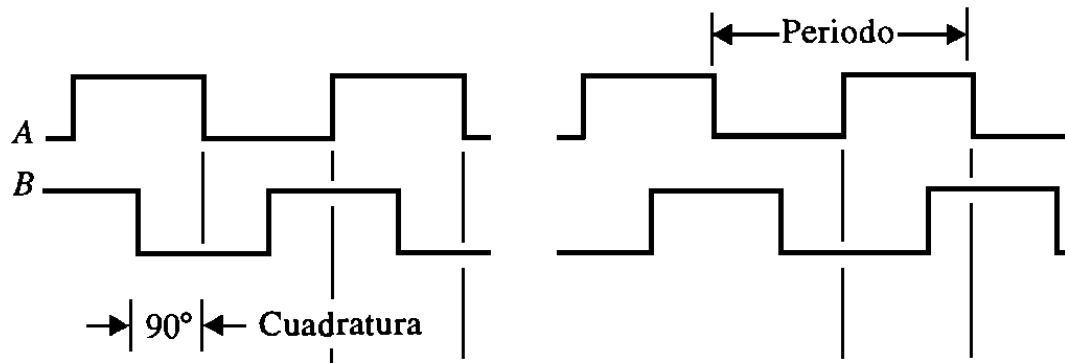


Figura 18. Señal de un codificador de dos canales en cuadratura.

Fuente: Sistemas de Control Automático, Kuo, 1996.

### 1.23 PUENTE H

El puente H es un circuito electrónico que permite a un motor eléctrico DC girar en ambos sentidos, avanzar y retroceder. Los puentes H ya vienen hechos en algunos circuitos integrados, como el L298 o DRV8833, pero también se pueden construir a partir de componentes eléctricos y/o electrónicos.

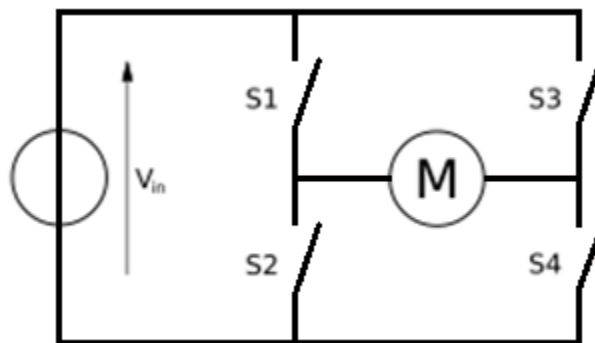


Figura 19. Puente H.

Fuente: Puente h para control de motores,

<http://www.ingmecafenix.com/electronica/puente-h-control-motores/>

Un puente H se construye con 4 interruptores (mecánicos o mediante transistores) como se expone en la figura 19. Cuando los interruptores S1 y S4 están cerrados (S2 y S3 abiertos) se aplica una tensión haciendo girar el motor en un sentido. Abriendo los interruptores S1 y S4 (cerrando S2 y S3), el voltaje se invierte, permitiendo el giro en sentido inverso del motor.

Un puente H no solo se usa para invertir el giro de un motor, también se puede usar para frenarlo de manera brusca, al hacer un corto entre los bornes del motor, o incluso puede usarse para permitir que el motor frene bajo su propia inercia, cuando desconectamos el motor de la fuente que lo alimenta.

La forma más común de hacer un puente H es usando transistores, puesto que sus tiempos de vida y frecuencias de conmutación son mucho más altos. Además los transistores se acompañan de diodos que permitan a las corrientes circular en sentido inverso al previsto cada vez que se conmute la tensión, puesto que el motor está compuesto por bobinados que durante varios períodos de tiempo se opondrán a que la corriente varíe (Puente h para control de motores).

## CAPÍTULO 2: ANÁLISIS Y PRESENTACIÓN DE RESULTADOS

### 2.1 ANÁLISIS DEL TRABAJO

Se realizó una entrevista<sup>4</sup> al responsable de laboratorio de Automatización para corroborar la utilidad de este proyecto, en dicha entrevista el Ing. Bayardo Juárez nos indica sobre la falta de prácticas de laboratorio en donde se estudien temas como: modelado de sistemas, identificación de parámetros de sistemas, compensadores PID y LQR, a su vez nos comentaba que en su opinión el laboratorio no cuenta con las suficientes herramientas para realizar prácticas más robustas. Una vez se le hizo saber qué es un péndulo invertido y el objetivo de éste como herramienta de trabajo del laboratorio nos indicó que realmente sería una herramienta muy útil y que espera que los profesores la tomen en cuenta para utilizarla en sus clases, además agregó que el hecho de ser algo diseñado por los mismos estudiantes enriquece y favorece mucho a la universidad.

De igual manera se entrevistó<sup>4</sup> al MBA Ing. Francisco Galo, él nos comenta que en su opinión el laboratorio de Automatización debería de irse actualizando de manera periódica, pero que muchas veces el factor dinero es un problema, así que le parece genial la idea de que los mismos estudiantes donen sus trabajos a la universidad, pues además de no tener que invertir en equipos, los estudiantes de las nuevas generaciones contarán con mejores herramientas para realizar sus prácticas de laboratorio.

Además de las entrevistas se han tomado algunas fotos<sup>5</sup> a los equipos con los que cuenta el laboratorio de Automatización para corroborar que hacen falta herramientas de aprendizaje de temas como modelado matemático, identificación de parámetros, diseño e implementación de compensadores PID y LQR.

---

<sup>4</sup> Ver en anexo 1.

<sup>5</sup> Ver en anexo 2.

## 2.2 CONSTRUCCIÓN DEL PROTOTIPO

La construcción parte de una impresora, siendo ésta el marco principal de todo el prototipo. Se utilizó una impresora HP Deskjet 840c que se expone en la figura 20. En realidad, se puede utilizar cualquier impresora, pero ésta es la que se tenía al alcance.

Por motivos de estética y presentación, se desarmó la impresora por completo y se pintó el marco de color negro como se muestra en la figura 21. Se ensamblaron ciertos componentes que ya trae la impresora, los cuales se muestran en la figura 22, que son el riel, el carro, la correa y otros aditamentos necesarios para su correcta fijación al marco.



Figura 20. Impresora HP Deskjet 840c.

Fuente: [www.hp.com](http://www.hp.com)



Figura 21. Marco de la impresora.

Fuente: Autor.

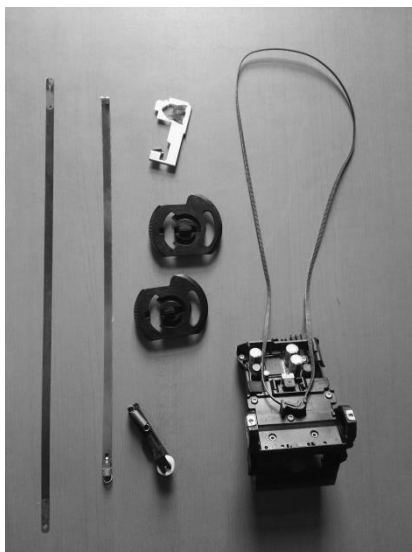


Figura 22. Partes de la impresora.

Fuente: Autor.

Al marco se le agregó el codificador incremental rotatorio del péndulo marca YUMO<sup>6</sup> modelo A6B2-CWZ3E-1024 de 1024 pulsos por revolución, que se muestra en la figura 23, y el motor<sup>7</sup> de engranajes de cc que se muestra en la figura 24 que trae su encoder de 374 pulsos por revolución unido al eje del mismo. Ambos encoders o codificadores son los responsables de medir tanto el desplazamiento angular del péndulo como el desplazamiento traslacional del carro respectivamente.



Figura 23. Codificador incremental rotatorio YUMO A6B2-CWZ3E-1024

Fuente: [www.sparkfun.com](http://www.sparkfun.com).

---

<sup>6</sup> Ver datasheet en anexo 3.

<sup>7</sup> Ver datasheet en anexo 4.



Figura 24. Motor de engranaje de CC con el codificador de Hall.

Fuente: [www.dx.com](http://www.dx.com).

Los componentes antes mencionados agregados al marco se muestran en sus vistas anterior y posterior de las figuras 25 y 26 respectivamente. El eje del motor de cc va atado al carro a través de la correa que traduce el movimiento rotacional del motor al movimiento traslacional del carro. De igual manera, el encoder rotatorio del péndulo va montado sobre el carro.

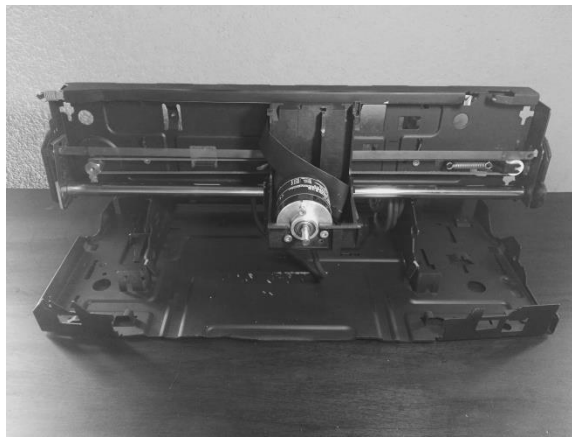


Figura 25. Vista anterior del marco de la impresora con codificador del péndulo y motor de engranajes de cc ensamblados.

Fuente: Autor.

La impresora trae su motor y su encoder que hacen posible que el carro que carga los cartuchos de tinta se muevan. Se decidió cambiar el motor y utilizar uno con engranajes y con su propio codificador en busca de un mayor control sobre el movimiento del carro, porque los engranajes reducen el

desplazamiento rotacional del eje del motor y por ende su velocidad, a expensas de menor robustez ante perturbaciones de gran amplitud sobre el péndulo.

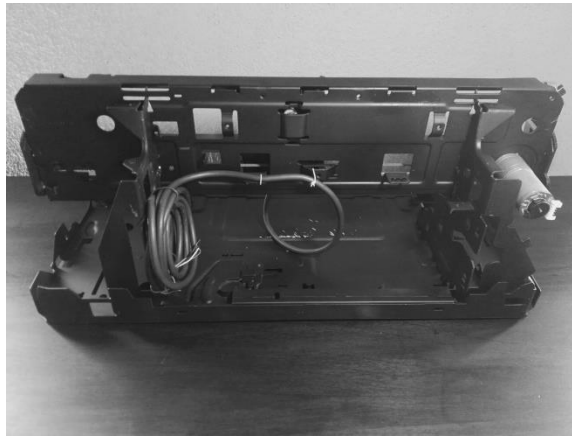


Figura 26. Vista posterior de marco de la impresora con codificador del péndulo y motor de engranajes de cc ensamblados.

Fuente: Autor.

Se utilizó el puente H L298N<sup>8</sup>, que se muestra en la figura 27, para brindarle la suficiente potencia de operación al motor.

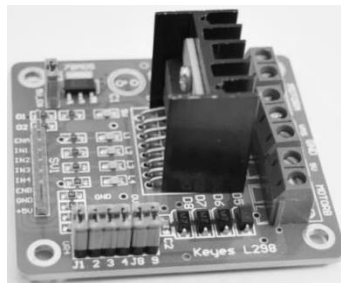


Figura 27. Puente H L298N.

Fuente: Autor.

Se construyó una tarjeta (figuras 28 y 29) personalizada y apilada sobre el Arduino Mega 2560 utilizando una tarjeta perforada, que contiene dos botones para el control manual del carro y un botón para el reset de la tarjeta. Se eligió el Arduino Mega 2560 sobre las otras tarjetas Arduino, debido a que posee una

---

<sup>8</sup> Ver datasheet en anexo 5.



mayor cantidad de interrupciones, ya que se necesitan 4 interrupciones, 2 para cada encoder. Se elaboró un manual de usuario<sup>9</sup> que explica funcionamiento del prototipo, la utilización los programas empleados en los acápites subsiguientes, así como las conexiones físicas de los componentes.

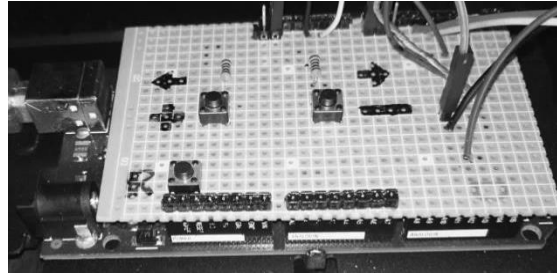


Figura 28. Tarjeta personalizada apilable.

Fuente: Autor.

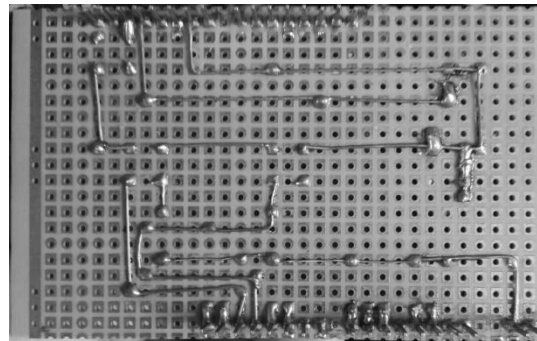


Figura 29. Parte inferior de la tarjeta personalizada apilable.

Fuente Autor.

La varilla utilizada para el péndulo es un tubo de aluminio de 8 mm de diámetro, el cual se dobló cerca de un extremo con una curvadora o dobladora manual de tubos formando un ángulo de 90°. En el extremo doblado, se le colocó un tornillo el cual se ajusta para unirlo al eje del codificador incremental YUMO que está sobre el carro. El prototipo del péndulo sobre el carro con todos los componentes<sup>10</sup> ensamblados se presenta en las figuras 31 y 32.

---

<sup>9</sup> Ver anexo 7.

<sup>10</sup> Ver lista de partes en anexo 8.

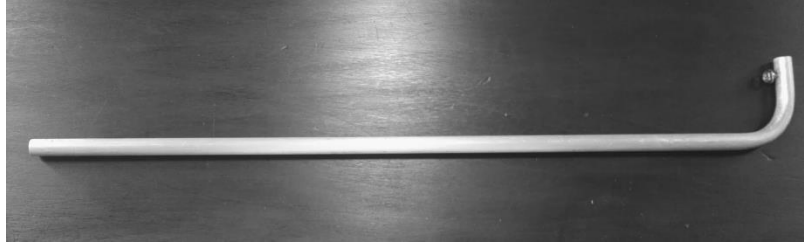


Figura 30. Varilla de aluminio del péndulo.

Fuente: Autor.

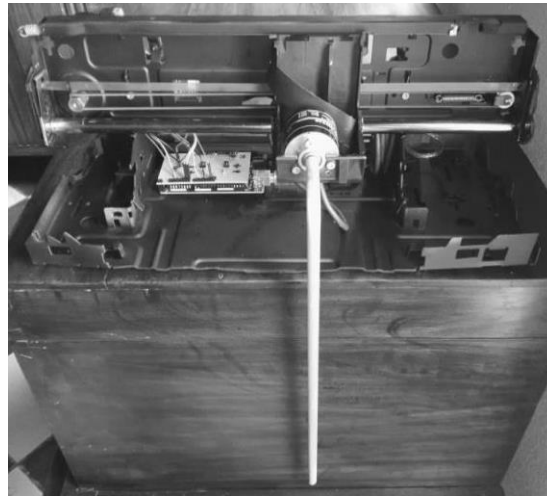


Figura 31. Vista anterior del prototipo con todos los componentes ensamblados.

Fuente: Autor.

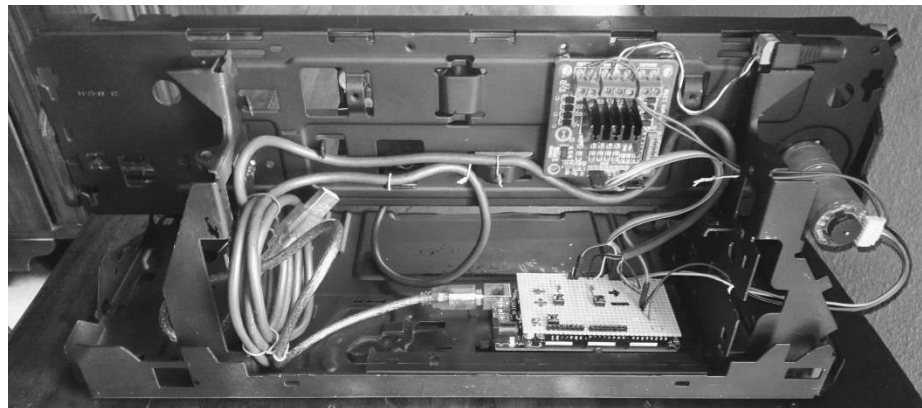


Figura 32. Vista posterior del prototipo con todos los componentes ensamblados.

Fuente: Autor.

## 2.3 IDENTIFICACIÓN DE LOS PARÁMETROS DEL PÉNDULO

La manera más fácil de estudiar el péndulo es en su configuración estable colgando hacia abajo. Para ello se utilizó la ecuación (56) linealizada para ángulos pequeños, la cual se expone a continuación

$$\ddot{\theta} + \frac{b}{ml^2}\dot{\theta} + \frac{g}{l}\theta = 0 \quad (65)$$

Lo importante a saber de esta ecuación es su forma, ya que puede representar un sistema en un dominio diferente, por ejemplo un circuito RLC, o un sistema masa, resorte, amortiguador. Por lo tanto, no se abordó la identificación de los parámetros del péndulo  $g, l, b$  y  $m$  realizando mediciones físicas sobre éste, o utilizando la ya conocida constante de gravedad; sino que se trató al sistema como una señal que puede presentarse en sistemas de dominio distintos y cuya principal diferencia entre estos sistemas es la interpretación física de las ecuaciones. El modelo matemático general es de la forma de la ecuación (10) mostrada nuevamente a continuación

$$\ddot{x} + 2\alpha\dot{x} + \omega_0^2x = 0 \quad (66)$$

Así, los coeficientes de la ecuación linealizada del péndulo pasan a ser

$$2\alpha = \frac{b}{ml^2} \quad \omega_0^2 = \frac{g}{l} \quad (67)$$

Entonces, para obtener el modelo del péndulo o su ecuación diferencial, es necesario conocer la constante de amortiguamiento  $\alpha$  y la velocidad angular de oscilación no amortiguada  $\omega_0$ .

En la enseñanza de ecuaciones diferenciales es común ver problemas donde se brinda una ecuación diferencial ordinaria con sus condiciones iniciales para obtener su solución y su gráfica o señal si acaso es requerida. A este problema se le llama problema del valor inicial o problema de Cauchy. Sin embargo, en este caso se tuvo que ir en sentido inverso, a partir de una señal se obtuvo la ecuación diferencial.

Para ello se realizó un experimento en el péndulo con la configuración hacia abajo como en la figura 12, soltándolo desde un ángulo inicial  $\theta$  menor a diez grados ó 0.17 radianes con el objeto de que el modelo linealizado se mantuviera válido. La conversión es de 4096 pulsos del encoder a 360 grados o

$2\pi$  radianes. Se utilizó el programa *Pendulum\_identification\_embedded.slx* que corre en el Arduino Mega2560 y *Pendulum\_identification\_host.slx* que corre como host en el computador para la adquisición de los datos del experimento. Posterior, para el análisis de estos datos experimentales se utilizó el script de Matlab *Pendulum\_identification.m*<sup>11</sup> con lo que se obtuvo la figura 33 y todas figuras subsiguientes de este acápite 2.3.

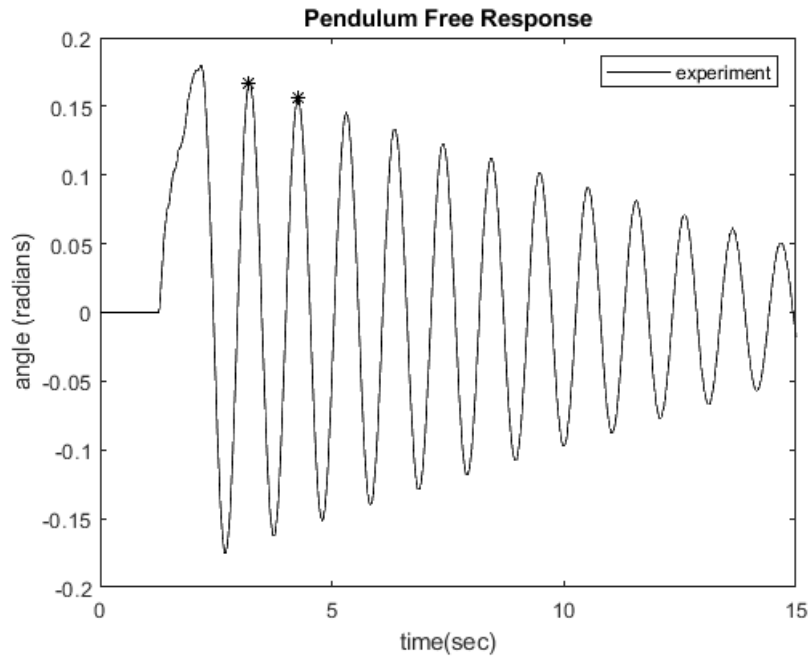


Figura 33. Experimento en el péndulo.

Fuente: Autor.

Como se puede observar, esta señal es una oscilación amortiguada y tiene la forma de la respuesta libre del sistema de segundo orden subamortiguado de la figura 3 (b). Esto quiere decir que el comportamiento del péndulo, modelado mediante un sistema de segundo orden subamortiguado, está completamente caracterizado por la constante de amortiguamiento  $\alpha$  y la velocidad angular amortiguada de oscilación  $\omega_d$ .

---

<sup>11</sup> Ver en anexo 6.

Para identificar  $\alpha$  y  $\omega_d$ , se utilizó las dos primeras máximas de la respuesta libre graficadas con un asterisco en la figura 33 y cuyas coordenadas son las siguientes

$$(t_1, y_1) = (3.21, 0.1672)$$

$$(t_2, y_2) = (4.26, 0.1565)$$

La ecuación (27) es de utilidad en este caso, ya que se puede ver con claridad que el sistema de segundo orden subamortiguado está gobernado por una exponencial de raíz real negativa que es la envolvente, y un coseno desfasado. La multiplicación de ambos da como resultado una senoide amortiguada, que es la respuesta libre del sistema. Suponiendo que la exponencial de la envolvente pasa por las dos máximas y que está desplazada en  $t_1$  de la siguiente manera

$$y(t) = y_1 e^{-\alpha(t-t_1)} \quad (68)$$

se evaluó la ecuación (68) en  $y(t_2)$

$$y_2 = y_1 e^{-\alpha(t_2-t_1)} \quad (69)$$

así se obtuvo la ecuación para la constante de amortiguamiento

$$\alpha = -\frac{1}{(t_2 - t_1)} \ln \frac{y_2}{y_1} \quad (70)$$

sustituyendo los valores de  $(t_1, y_1)$  y  $(t_2, y_2)$  en la ecuación (70)

$$\alpha = -\frac{1}{(4.26 - 3.21)} \ln \frac{0.1565}{0.1672}$$

se encontró el valor numérico de la constante de amortiguamiento del péndulo

$$\alpha = 0.0632 \frac{1}{\text{segundos}} \quad (71)$$

Para la velocidad angular amortiguada de oscilación, se utilizó la ecuación (29) que se muestra a continuación

$$T = \frac{2\pi}{\omega_d} \quad (72)$$

obteniendo

$$\omega_d = \frac{2\pi}{t_2 - t_1} = 5.984 \frac{\text{radianes}}{\text{segundos}} \quad (73)$$

Es importante señalar que los parámetros  $\alpha$  y  $\omega_d$  no están en unidades de radianes, ni grados, ni pulsos del encoder. Por lo tanto, no importa en qué unidades esté la señal utilizada en la identificación, ya que estos parámetros no están en función de las unidades de la señal, sino del tiempo.

Utilizando la ecuación (66)

$$\ddot{\theta} + 2\alpha\dot{\theta} + \omega_0^2\theta = 0 \quad (74)$$

sustituyendo  $\omega_0^2$  de la ecuación (19)

$$\ddot{\theta} + 2\alpha\dot{\theta} + (\alpha^2 + \omega_d^2)\theta = 0$$

posteriormente los valores de  $\alpha$  y  $\omega_d$

$$\ddot{\theta} + 0.1264\dot{\theta} + 35.81\theta = 0 \quad (75)$$

Se obtiene la ecuación homogénea del péndulo (75) con los parámetros identificados y cuyas raíces son complejas conjugadas, las cuales se obtienen mediante la ecuación (20)

$$s_{1,2} = -0.0632 \pm j5.984 \quad (76)$$

También se puede apreciar que se cumple la condición  $0 < \alpha < \omega_0$ , o bien,  $0 < 0.0632 < 5.9843$  para un sistema de segundo orden subamortiguado.

Para validar el modelo de la ecuación (75), se simuló y graficó comparándolo con los datos experimentales de la respuesta libre del péndulo de la figura 33, obteniendo la figura 34.

Como se puede observar, la gráfica del modelo que se manifiesta mediante leyenda “model” en la figura 34, describe lo suficientemente bien la dinámica del péndulo. Esta gráfica del modelo se obtuvo mediante la solución al problema del valor inicial o problema de Cauchy de la ecuación diferencial (75), cuyas condiciones iniciales se tomaron de la primer máxima  $\theta(0) = y_1 = 0.1672$  *radianes* y  $\dot{\theta}(0) = 0$   $\frac{\text{radianes}}{\text{segundo}}$ .

Para hallar la solución analítica a la ecuación (75) se utilizó la ecuación (13) que se muestra a continuación

$$\theta(t) = K_1 e^{s_1 t} + K_2 e^{s_2 t} \quad (77)$$

Las raíces  $s_{1,2}$  ya han sido encontradas en la ecuación (76). Para encontrar los valores de  $K_1$  y  $K_2$  se utilizaron las ecuaciones (16) y (17), que se presentan subsiguientemente en función de  $\theta$

$$K_1 = \frac{\dot{\theta}(0) - s_2\theta(0)}{s_1 - s_2} \quad (78)$$

y

$$K_2 = \frac{\dot{\theta}(0) - s_1\theta(0)}{s_2 - s_1} \quad (79)$$

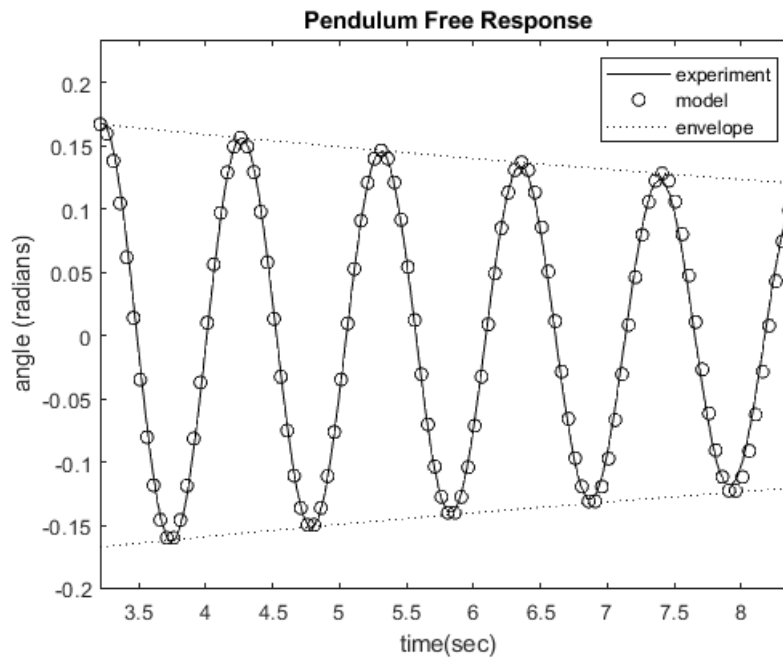


Figura 34. Validación del modelo del péndulo.

Fuente: Autor.

Sustituyendo los valores de  $\theta(0)$ ,  $\dot{\theta}(0)$ ,  $s_1$  y  $s_2$  en las ecuaciones (78) y (79) obtenemos

$$K_1 = 0.0836 - j0.0009 \quad (80)$$

y

$$K_2 = 0.0836 + j0.0009 \quad (81)$$

con lo que se ratifica que estas constantes son complejas conjugadas  $K_1 = \overline{K_2}$ . Luego queda la solución (77)

$$\begin{aligned}\theta(t) = & (0.0836 - j0.0009)e^{(-0.0632+j5.984)t} \\ & + (0.0836 + j0.0009)e^{(-0.0632-j5.984)t}\end{aligned}\quad (82)$$

En esta ecuación (82) no se distingue la senoide amortiguada. Por ésta razón se la representó en la forma de la ecuación (27), con el desfase  $\beta$  de la ecuación (28) en radianes

$$\theta(t) = 0.1672 e^{-0.06321 t} \cos(5.984t + 0.01056) \quad (83)$$

Posteriormente, se utilizó el plano complejo para graficar las raíces del péndulo de la ecuación (76) como se expone en la figura 35. Estas raíces están en el semiplano complejo izquierdo, lo que significa que su parte real es negativa originando exponenciales negativas como en la ecuación (83), cuya amplitud decrece con el tiempo, dando como resultado un sistema estable.

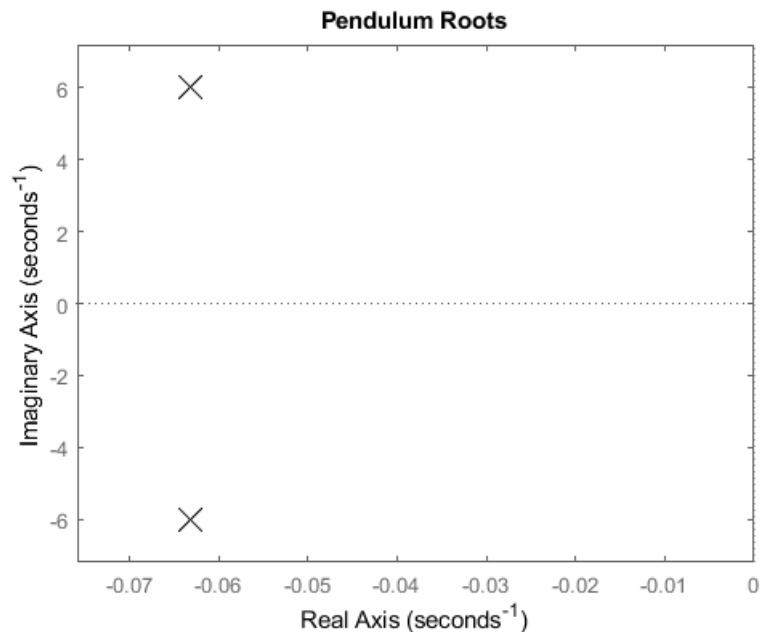


Figura 35. Raíces del péndulo en el plano complejo  $s$ .

Fuente: Autor.



## 2.4 IDENTIFICACIÓN DE LOS PARÁMETROS DEL CARRO

Como se mencionó anteriormente en la sección 1.14, el estudio del carro es esencialmente el estudio del motor, esto debido a que el desplazamiento angular del eje del motor se traduce al movimiento traslacional del carro mediante una correa. Esta traducción de un movimiento a otro está descrita matemáticamente mediante la relación lineal de la fórmula de la longitud del arco.

El modelo matemático del motor dc está dado por la ecuación (52), que se presenta a continuación.

$$T_m \ddot{\phi} + \dot{\phi} = K_m e_a \quad (84)$$

La función de transferencia (51) expone con mayor claridad la relación que existe entre la posición angular del eje del motor y el voltaje aplicado al mismo, siendo la de un sistema de primer orden por un integrador. Si la relación fuese entre la velocidad angular del eje del motor y el voltaje aplicado, solamente sería un sistema de primer orden, sin el integrador. Y es que para pasar de la posición, a la velocidad y luego a la aceleración, se deriva. Y en sentido inverso, se integra.

De la misma manera que en el caso del péndulo, lo importante de la ecuación (84), es su forma. Se realizó un cambio de variables y se dividió cada uno de los términos de la ecuación (84) entre  $K_m$  ajustándola a la ecuación (30) quedando de la siguiente manera

$$p_2 \ddot{y} + p_1 \dot{y} = u \quad (85)$$

así los coeficientes pasan a ser

$$p_2 = \frac{T_m}{K_m}, \quad p_1 = \frac{1}{K_m} \quad (86)$$

Se realizó un experimento sobre el carro moviéndolo de un extremo a otro para capturar su dinámica. Se utilizó el programa de Simulink *Dc\_motor\_identification\_embedded.slx* que corre en el Arduino Mega2560, y el programa *Dc\_motor\_identification\_host.slx* que corre como host en el computador para la adquisición de los datos experimentales utilizados en la identificación del sistema. Posteriormente, se utilizó el script en Matlab *Dc\_motor\_identification.m*<sup>11</sup> para la

obtención del modelo del motor que equivale al modelo del carro, y de las figuras subsiguientes de este acápite 2.4.

En la figura 36 se observa que la señal de entrada al motor utilizada es una señal pwm constante a 255 con cierta duración debido a que riel sobre el cual se mueve el carro tiene una distancia finita. A pesar de esta limitante, el experimento es capaz de captar el transitorio y el estado estable del sistema, siendo el transitorio la curva que se presenta al inicio del cambio de la posición del eje del motor cuando la señal pwm pasa de 0 a 255, y el estado estable se presenta en la pendiente constante de la posición, lo que significa una velocidad constante. También se manifiesta otro transitorio en la posición del eje del motor debido a la energía almacenada en el sistema, cuando la señal pwm pasa de 255 a 0.

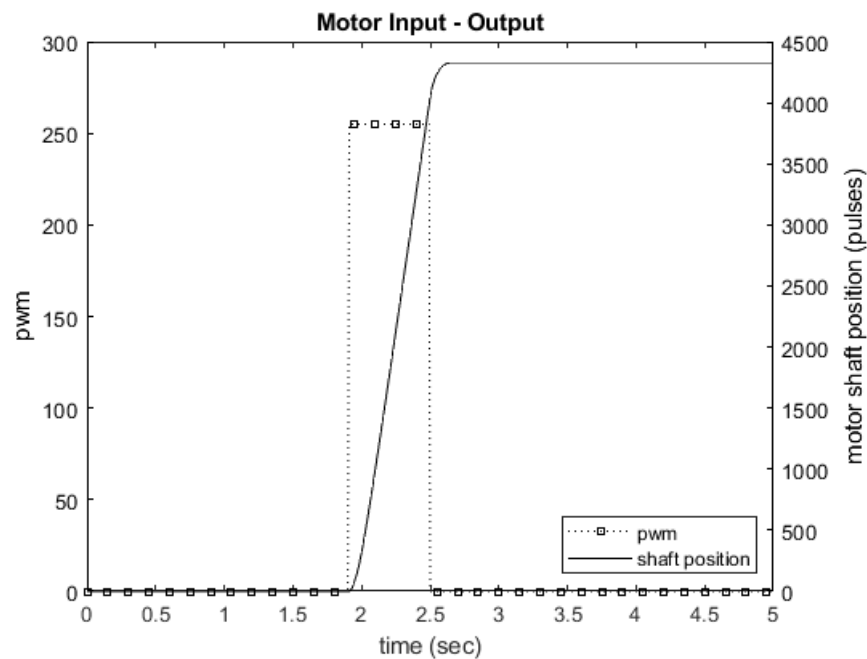


Figura 36. Datos experimentales de entrada-salida del motor dc.

Fuente: Autor.

En la identificación de los coeficientes de la ecuación (85) se realizó un arreglo matricial como el que se muestra en la ecuación (32) con la primera y segunda integración numérica de la posición del eje del motor  $y$ , y la segunda integración numérica de la entrada pwm  $u$ . Para resolver este sistema de

ecuaciones lineales sobredeterminado  $Ax = B$  se utilizó el backslash operator en Matlab, el cual devuelve una solución de mínimos cuadrados al sistema de ecuaciones. Ésta solución es el vector  $p$  con los coeficientes  $p_2$  y  $p_1$  de la ecuación diferencial (85).

Así se obtuvo

$$p_2 = 0.001864, \quad p_1 = 0.03474 \quad (87)$$

o bien

$$T_m = 0.0537, \quad K_m = 28.7893 \quad (88)$$

Con estos datos las ecuaciones (84) y (85) quedan

$$0.0537 \ddot{\phi} + \dot{\phi} = 28.7893 e_a \quad (89)$$

$$0.001864 \ddot{y} + 0.03474 \dot{y} = u \quad (90)$$

y la función de transferencia del motor dc basada en la ecuación (89)

$$M(s) = \frac{28.7893}{s(0.0537s + 1)} \frac{\text{pulses}}{\text{pwm}} \quad (91)$$

siendo las unidades de la posición del motor en pulsos del encoder y de la entrada en unidades de pwm.

Con el objeto de validar el modelo de la función de transferencia (91) se simuló excitándolo con la misma entrada del experimento de la figura 36, y la salida se le comparó con la del mismo experimento, con lo que se obtuvo la figura 37. Como se puede observar, el modelo describe con fidelidad la dinámica del motor dc, y por ende, la dinámica del carro.

Las raíces del motor en base a la función de transferencia (91) se exponen en la figura 38. Una raíz se encuentra en el origen que es el integrador, y la otra se encuentra en el semiplano complejo izquierdo, la cual es el inverso de la constante de tiempo  $\frac{1}{T_m}$  y es real pura

$$\begin{aligned} s_1 &= 0 \\ s_2 &= 18.6314 \end{aligned} \quad (92)$$

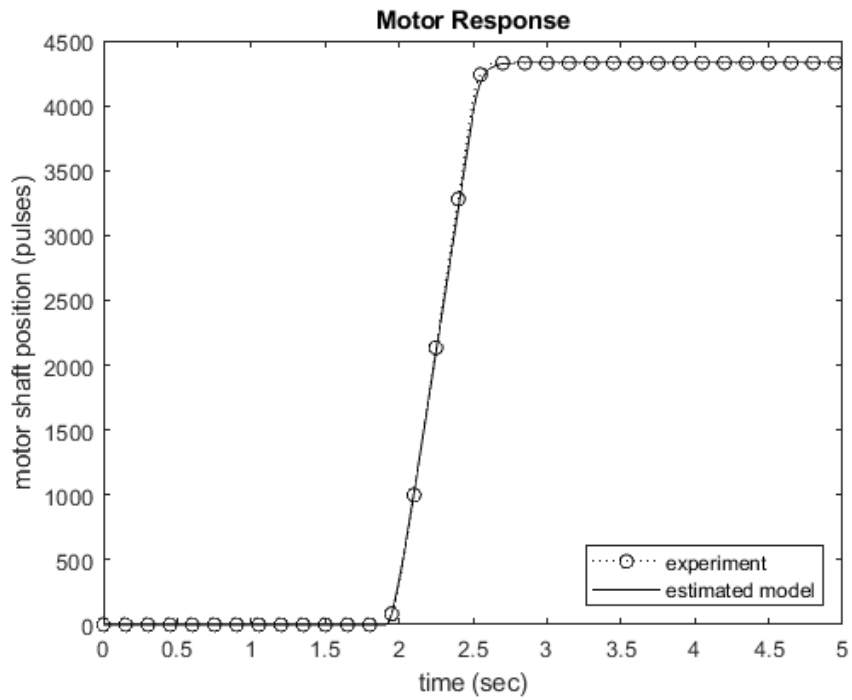


Figura 37. Validación del modelo del motor dc.

Fuente: Autor.

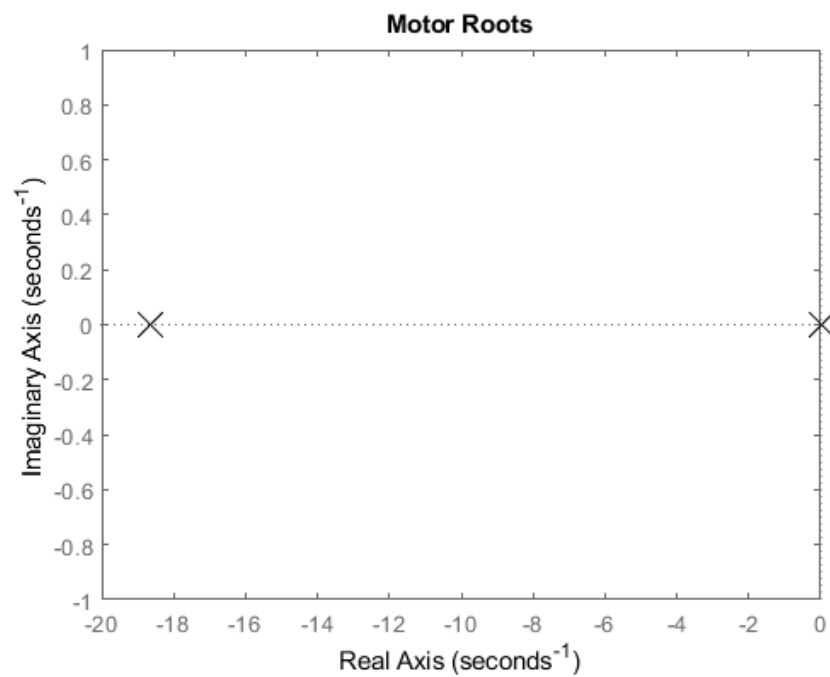


Figura 38. Raíces del motor en el plano complejo s.

Fuente: Autor.

## 2.5 IDENTIFICACIÓN DE LOS PARÁMETROS DEL PÉNDULO SOBRE EL CARRO

La identificación del sistema del péndulo sobre el carro está basada en las ecuaciones (41) y (42), las cuales se linealizan utilizando las aproximaciones  $\sin \theta \approx \theta$ ,  $\cos \theta \approx 1$  y  $\dot{\theta}^2 \approx 0$  dando como resultado las ecuaciones siguientes

$$(M + m)\ddot{x} + ml\ddot{\theta} = u \quad (93)$$

$$\ddot{\theta} + \frac{1}{l}\ddot{x} + \frac{g}{l}\theta = 0 \quad (94)$$

Con un poco de intuición, se analizan estas ecuaciones. La (93) es la ecuación del carro y es similar a las ecuaciones (84) y (85), con la diferencia que no tiene el término debido a la fricción  $\dot{x}$ , pero tiene un término  $\ddot{\theta}$  de más, que al trasladarlo a la derecha del igual equivale a una excitación al carro, lo que indica que el movimiento del carro está influenciado por el movimiento del péndulo. De la misma manera, la ecuación (94) es la ecuación del péndulo y es similar a la ecuación (65), con la diferencia que la ecuación (94) no tiene el término  $\dot{\theta}$  debido a la fricción, pero sí tiene un término  $\ddot{x}$  de más, que si se traslada a la derecha del igual equivale a una excitación, por lo que se puede interpretar que el movimiento del péndulo se debe al movimiento del carro.

Para la identificación de los parámetros del sistema del péndulo sobre el carro se realizó un experimento en el sistema físico real en la configuración de la figura 9. Se utilizó el programa de Simulink *Cart\_Pendulum\_identification\_embedded.slx* que corre en el Arduino Mega 2560, y el programa *Cart\_Pendulum\_identification\_host.slx* que corre como host en el computador, para la adquisición de los datos experimentales que luego son tratados en Matlab mediante el script *Cart\_pendulum\_identification.m*<sup>11</sup> con lo que se obtiene el modelo del sistema y de las figuras presentadas a lo largo de este acápite 2.5.

El experimento consistió en excitar al motor con una señal similar a la utilizada en la identificación del mismo en el acápite anterior, la cual provocó que el carro se moviera de un lado al otro y al mismo tiempo, que el péndulo oscilara. Estas oscilaciones son menores a 10 grados o 113 pulsos del encoder, por lo que son adecuadas para la identificación de las ecuaciones lineales del péndulo.

Los datos experimentales se muestran en la figura 39. Es importante notar a partir de la gráfica lo siguiente: primero, que mientras el péndulo oscila, la posición del carro no oscila, lo que quiere decir que no hay influencia del movimiento del péndulo sobre el carro; y por lo tanto, el término  $\ddot{\theta}$  en la ecuación (93) es nulo. Segundo, debido a que la posición del carro no oscila mientras el péndulo oscila, y suponiendo que no hay otras perturbaciones sobre el péndulo como por ejemplo el viento, estas oscilaciones libres son adecuadas para identificar los parámetros del péndulo como se realizó en el acápite 2.3.

Con lo dicho anteriormente, las ecuaciones (93) y (94) en combinación con las ecuaciones (84) y (74), quedan de la siguiente manera

$$T_m \ddot{x} + \dot{x} = K_m u \quad (95)$$

$$\ddot{\theta} + 2\alpha\dot{\theta} + \omega_0^2\theta = -a\ddot{x} \quad (96)$$

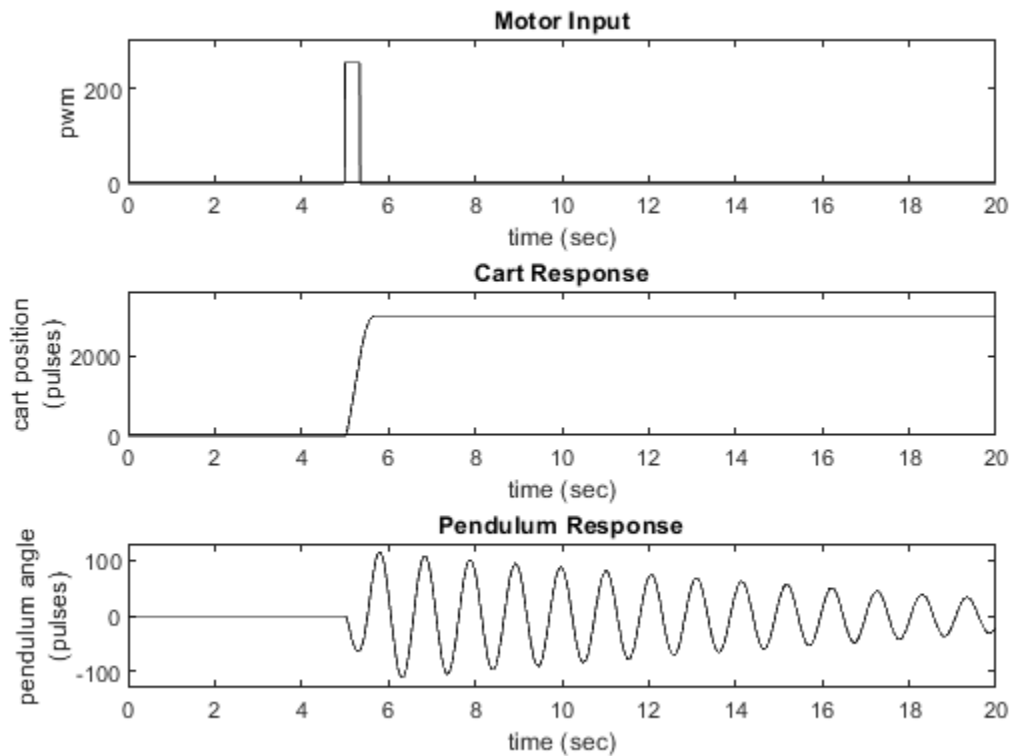


Figura 39. Datos experimentales de entrada-salidas del péndulo sobre el carro.

Fuente: Autor.

Entonces, la ecuación (95), nos dice que el sistema del carro está descrito por los parámetros  $T_m$  y  $K_m$ , que son la constante de tiempo y la ganancia, respectivamente. Ahora, la ecuación (96) nos dice que el péndulo está descrito completamente por los parámetros  $\omega_0^2$  y  $\alpha$ , que indican cuán rápido oscila y cuán rápido decaen las oscilaciones. Pero, aún falta el coeficiente  $a$  de la ecuación (96), que prácticamente actúa como una ganancia, así como lo hace la ganancia  $K_m$ . Este coeficiente  $a$  es una constante que se ha incluido por conveniencia, que puede ser interpretada como una ganancia, un parámetro de conversión o un constante de proporcionalidad que afecta la amplitud de las oscilaciones del péndulo. En este parámetro está incluido tanto el coeficiente  $\frac{1}{l}$  de la ecuación (94), como la constante de proporcionalidad  $r$  de la fórmula (43) de la longitud del arco, el cual relaciona el desplazamiento rotacional del eje de motor  $\phi$ , con el desplazamiento traslacional del carro  $x$ .

Con el objeto de simular el modelo dinámico del sistema lineal de las ecuaciones (95) y (96), se le representó en términos de la ecuación diferencial de vectores de estado (59), quedando de la siguiente manera

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{T_m} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{a}{T_m} & -\omega_0^2 & -2\alpha \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_m}{T_m} \\ 0 \\ -\frac{a K_m}{T_m} \end{bmatrix} u \quad (97)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \quad (98)$$

Utilizando los métodos ya descritos en las identificaciones independientes del péndulo y del carro de los acápites anteriores, se estimaron los parámetros

$$K_m = 34.0061, \quad T_m = 0.1124, \quad \alpha = 0.063822, \quad \omega_0 = 6.0419 \quad (99)$$

El único parámetro desconocido es la constante  $\alpha$ , la cual se asumió con un valor inicial de la unidad. Se simuló el sistema de las ecuaciones de estado (97) y (98), utilizando el mismo script en Matlab *Cart\_Pendulum\_identification.m*<sup>11</sup>, con los parámetros estimados (99) generando la figura 40.

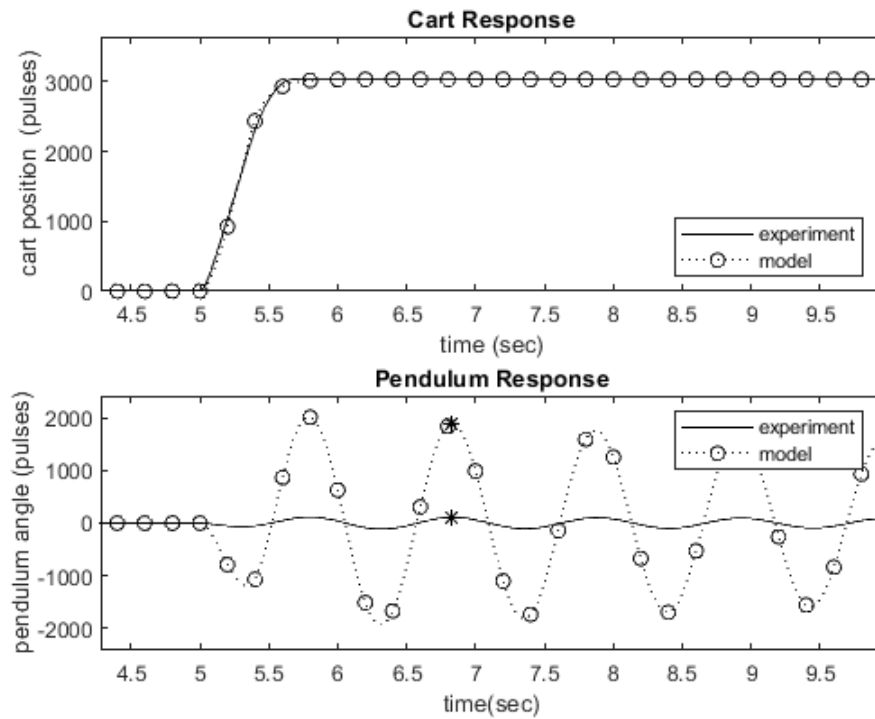


Figura 40. Validación del modelo del péndulo sobre el carro con  $\alpha = 1$ .

Fuente: Autor.

Se puede observar que el modelo del carro se ajusta bien a los datos experimentales, pero no así el modelo del péndulo. En la gráfica de la respuesta del péndulo, se nota que el modelo captura la dinámica, pero difiere en amplitud a los datos experimentales, siendo mayor la amplitud del modelo a la del experimento. Esto se debe a que el coeficiente  $\alpha$  no es el adecuado. Por consiguiente, para ajustarlo se definió el parámetro  $\alpha$  como la razón de la



segunda máxima del experimento, entre la segunda máxima del modelo marcados con un asterisco en la figura 40, siendo

$$a = \frac{2^{da} \text{ máxima del experimento}}{2^{da} \text{ máxima del modelo}} = 0.0583 \quad (100)$$

Se simuló nuevamente el modelo con todos los coeficientes estimados y se obtuvo la figura 41.

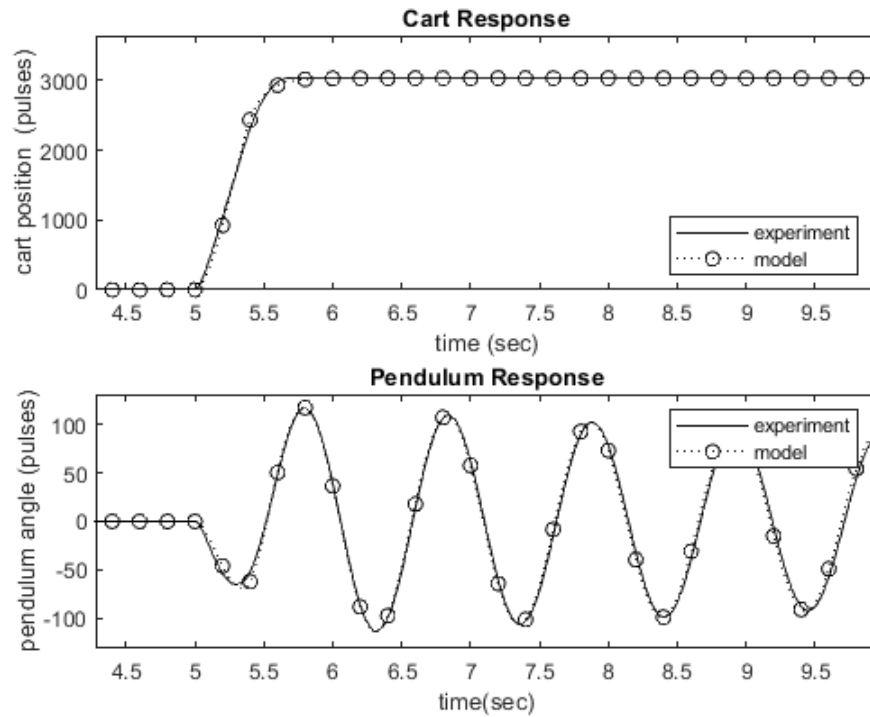


Figura 41. Validación del modelo con el coeficiente estimado  $a = 0.0577$ .

Fuente: Autor.

Así el modelo estimado del péndulo sobre el carro en espacio de estados es

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -8.899 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0.5186 & -36.5 & -0.1276 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 302.6 \\ 0 \\ -17.64 \end{bmatrix} u \quad (101)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \quad (102)$$

Es importante mencionar que no se realizó ninguna conversión de los pulsos de los encoders del carro y del péndulo, por lo que las unidades de los estados están en pulsos para las posiciones y en pulsos por segundo para las velocidades.

## 2.6 IDENTIFICACIÓN DE LOS PARÁMETROS DEL PÉNDULO INVERTIDO SOBRE EL CARRO.

Al comparar las ecuaciones (39) y (40) del péndulo invertido sobre el carro con las ecuaciones (93) y (94), la única diferencia es en el signo del término del torque debido a la gravedad; específicamente el término  $\theta$  en las ecuaciones (40) y (94) del péndulo. Esta diferencia en el signo es la que dicta si el sistema es estable o no.

Con lo dicho anteriormente, las ecuaciones del péndulo invertido sobre el carro basadas en las ecuaciones (95) y (96) son

$$T_m \ddot{x} + \dot{x} = K_m u \quad (103)$$

$$\ddot{\theta} + 2\alpha \dot{\theta} - \omega_0^2 \theta = -a \ddot{x} \quad (104)$$

Por consiguiente, el modelo en espacio de estados es

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{T_m} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{a}{T_m} & \omega_0^2 & -2\alpha \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_m}{T_m} \\ 0 \\ -\frac{a K_m}{T_m} \end{bmatrix} u \quad (105)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \quad (106)$$

Sustituyendo los parámetros estimados de las ecuaciones (99) y (100) el modelo del péndulo invertido sobre el carro queda de la siguiente manera

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -8.899 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0.5186 & 36.5 & -0.1276 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 302.6 \\ 0 \\ -17.64 \end{bmatrix} u \quad (107)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \quad (108)$$

La función de transferencia del carro queda

$$\frac{X(s)}{U(s)} = \frac{302.6}{s^2 + 8.899 s} \frac{\text{pulses}}{\text{pwm}} \quad (109)$$

Y el péndulo invertido sobre el carro

$$\frac{\Theta(s)}{U(s)} = \frac{-17.64 s^2}{s^4 + 9.026 s^3 - 35.37 s^2 - 324.8 s} \frac{\text{pulses}}{\text{pwm}} \quad (110)$$

Siendo la entrada de la ecuación (110) la señal pwm  $u$ , y la salida el ángulo  $\theta$  del péndulo invertido en unidades de pulsos del encoder.

Ahora, la respuesta al impulso en lazo abierto se presenta en la figura 42. Se puede observar que el sistema es inestable, ya que el ángulo del péndulo  $\theta$  crece ilimitadamente. Esto también se puede confirmar mediante la ubicación de las raíces de la función de transferencia (110) presentadas a continuación

$$\begin{aligned} \text{Ceros} &= \begin{matrix} 0 \\ 0 \end{matrix} \\ \text{Polos} &= \begin{matrix} 0 \\ 5.978 \\ -8.898 \\ -6.106 \end{matrix} \end{aligned} \quad (111)$$

En donde al cancelar polos y ceros en común, queda un polo en 5.978 ubicado en el semiplano complejo derecho, teniendo la solución de la función de transferencia una exponencial con raíz positiva que aumenta sin límite a medida que el tiempo avanza desde el instante  $t = 0$ .

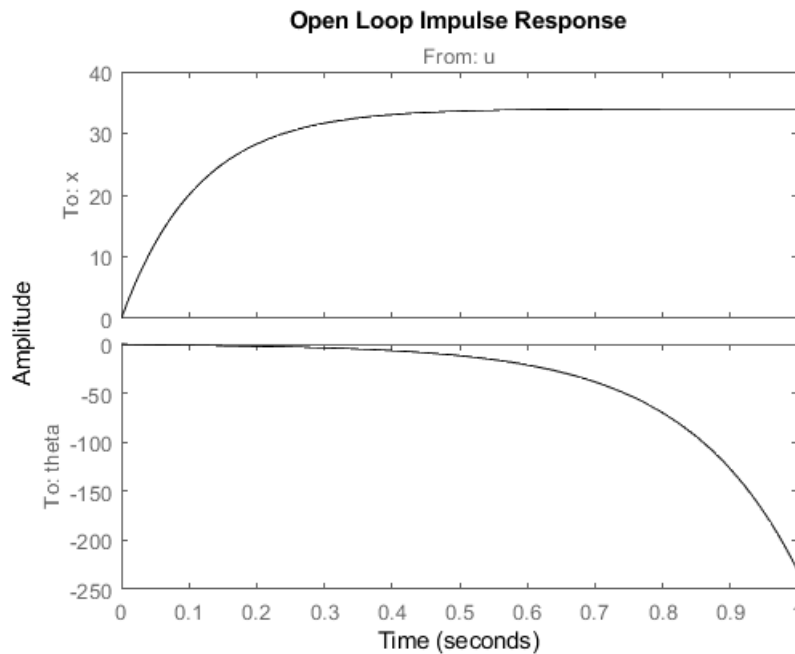


Figura 42. Respuesta al impulso en lazo abierto del sistema péndulo invertido sobre el carro.

Fuente: Autor.

## 2.7 DISEÑO DEL CONTROLADOR PID PARA EL PÉNDULO INVERTIDO SOBRE EL CARRO MEDIANTE EL LUGAR GEOMÉTRICO DE LAS RAÍCES

El diseño del controlador no es más que una pequeña parte de todo el proceso que conlleva el diseño del sistema de control, siendo ésta la que menor tiempo de realización requiere. El diseño del compensador PID tiene un poco de arte, pues es necesario cierta destreza del ingeniero de control en la selección de los parámetros o ubicación de las raíces del controlador.

El diagrama de bloques del sistema de control en lazo cerrado del péndulo invertido sobre el carro es el siguiente.

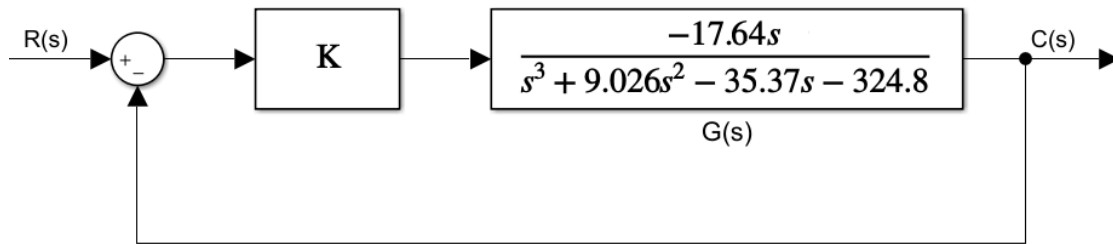


Figura 43. Sistema de control en lazo cerrado del péndulo invertido sobre el carro con controlador de ganancia proporcional  $K$ .

Fuente: Autor.

La planta  $G(s)$  está conformada por la función de transferencia (110) con los polos y ceros, que se muestran en la ecuación (111), cancelados. El controlador es una ganancia proporcional  $K$  y la realimentación es unitaria.

Se busca con la realimentación y la introducción del compensador en el lazo cerrado la estabilidad del sistema, o dicho de otra manera, ubicar los polos en lazo cerrado en el semiplano complejo izquierdo. Se diagrama el lugar geométrico de las raíces en la figura 44 utilizando el script de Matlab *Inverted\_Pendulum\_PID\_design*<sup>11</sup>. Todas las figuras subsiguientes en este acápite 2.7 también se obtuvieron mediante este script.

El comportamiento del lugar de las raíces sobre el eje real de la figura 44 no es el esperado, pues una de las reglas para el graficado indica que el lugar de las raíces puede existir en el eje real solamente a la izquierda de los polos o ceros reales de número impar; además, debe existir ahí. En otras palabras, los segmentos de los lugares de las raíces comienzan a la izquierda del primer polo o cero más a la derecha en el eje real, y se van alternando hacia la izquierda a lo largo del eje real.

Pero en este caso, el lugar de la raíz se encuentra a la derecha del primer polo ubicado más a la derecha sobre el eje real, y esto es debido al signo negativo del numerador de la función de transferencia  $G(s)$ , que hace actuar al lugar de la raíz como si fuese  $K < 0$ . También se puede interpretar como si la realimentación fuese positiva y no negativa. Esto afecta porque lo que se busca

es que el polo en el semiplano complejo derecho pase al semiplano complejo izquierdo con el objeto de estabilizar al sistema, pero lo que se observa es que al aumentar  $K$ , el polo en el semiplano complejo derecho busca el infinito positivo.

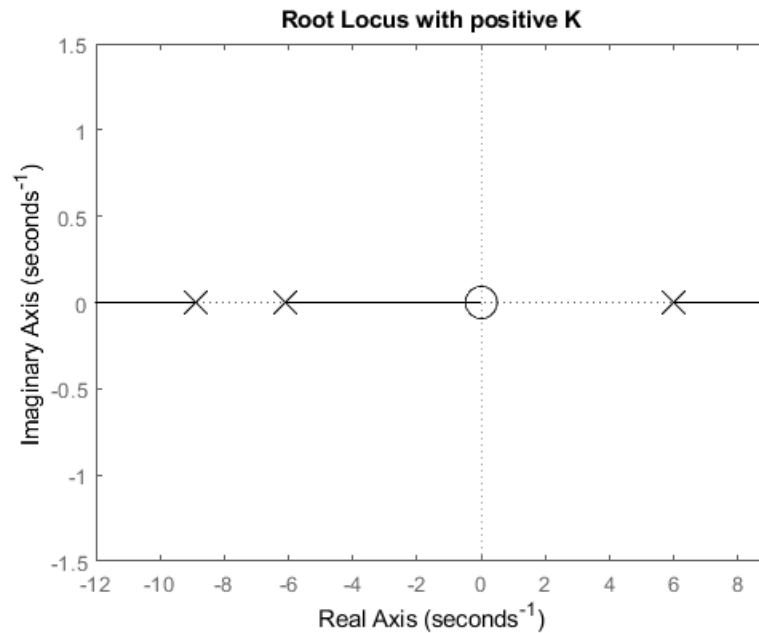


Figura 44. Lugar geométrico de las raíces del péndulo invertido sobre el carro.  
Fuente: Autor.

Por lo tanto, se elige una ganancia  $K$  negativa para cancelar el signo negativo del numerador de  $G(s)$ , como se muestra en la figura 45.

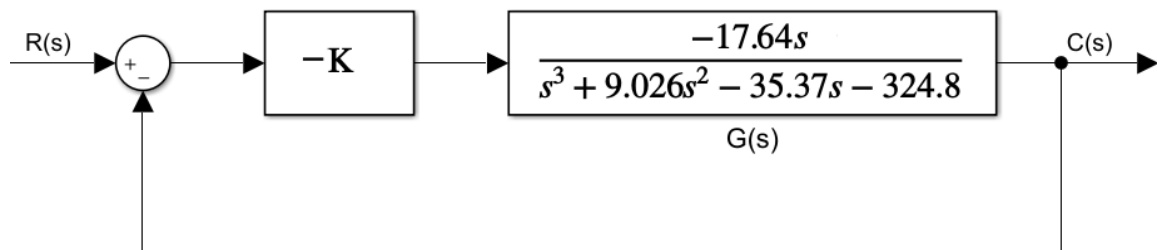


Figura 45. Sistema de control en lazo cerrado del péndulo invertido sobre el carro con controlador de ganancia proporcional  $-K$ .

Fuente: Autor.

Así el comportamiento del lugar de las raíces es como si fuese  $K \geq 0$  y  $G(s)$  positivo. Se graficó el lugar geométrico de las raíces para este sistema en la figura 46. Se puede observar a partir de esta figura, que cuando  $K \rightarrow +\infty$ , el polo en el semiplano complejo derecho tiende al cero en el origen, pero el sistema aún no es estable. Hasta el momento se ha avanzado en la dirección que toma el polo inestable.

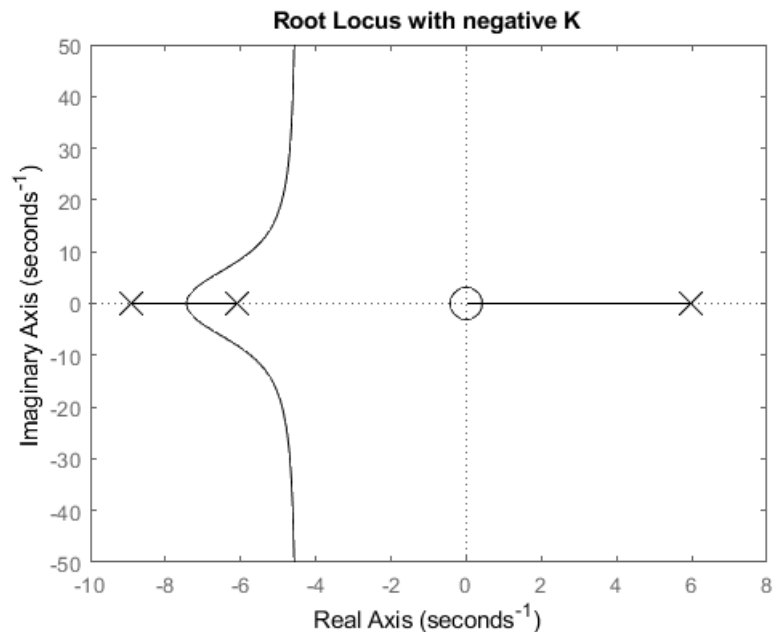


Figura 46. Lugar de las raíces con controlador de ganancia proporcional  $-K$ .

Fuente: Autor.

Para estabilizar el sistema, o en otras palabras, hacer que el polo inestable pase al semiplano complejo izquierdo, se diseñó un controlador PI, quedando el sistema en lazo cerrado como se expone en la figura 47. Se estableció el parámetro  $b = 3$ . El porqué de esta decisión radica en que el polo que agrega el compensador integral ideal cancela al cero en el origen, y la ganancia proporcional es la responsable de introducir un cero en  $-3$ , lo que provoca que el polo inestable pase al plano complejo izquierdo haciendo al sistema estable.

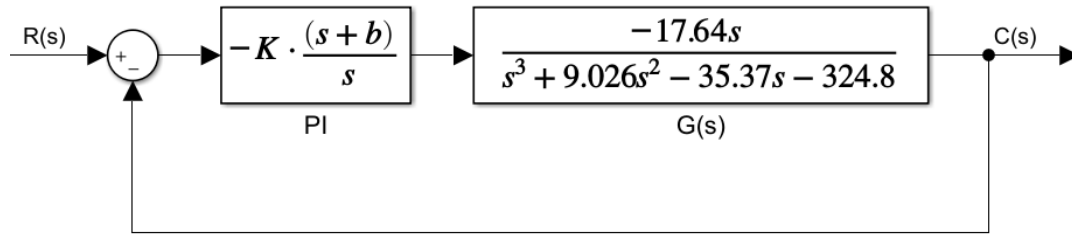


Figura 47. Controlador PI para el sistema del péndulo invertido sobre el carro.

Fuente: Autor.

El diagrama de las raíces para el sistema en lazo cerrado de la figura 48 se muestra a continuación.

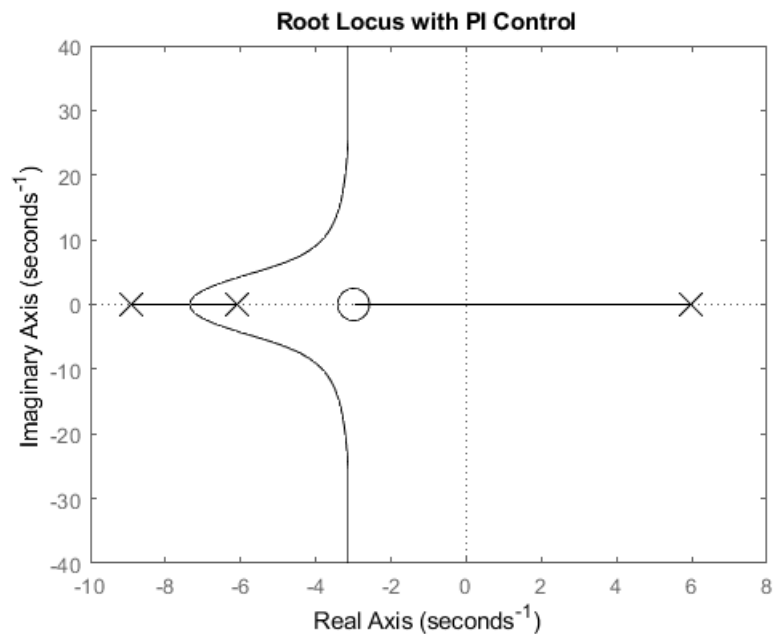


Figura 48. Lugar de las raíces con control PI y parámetro  $b = 3$ .

Fuente: Autor.

Aquí hay tres puntos importantes a destacar. Primero, la abscisa de intersección de las asíntotas y el eje real de los dos polos que se vuelven conjugados cuando  $K \rightarrow \infty$  y sus ángulos. Segundo, el punto de ruptura de salida sobre el eje real de estos polos conjugados. Tercero, el valor de  $K$  correspondiente al punto de cruce del polo inestable al semiplano complejo izquierdo que nos da la estabilidad.



La abscisa de intersección  $\sigma$  de las asíntotas y el eje real está dada por

$$\sigma = \frac{(\Sigma \text{ polos en lazo abierto}) - (\Sigma \text{ ceros en lazo abierto})}{p - z} \quad (112)$$

donde  $p$  es la cantidad de polos y  $z$  la cantidad de ceros en lazo abierto. Sustituyendo en la ecuación (112) los valores de los ceros y polos en lazo abierto del sistema de la figura 48

$$\sigma = \frac{(-8.899 - 6.106 + 5.978) - (-3)}{3 - 1} \approx -3.01 \quad (113)$$

Este valor de  $\sigma \approx -3.01$  indica que no importa cuán grande sea el valor de  $K$ , los polos conjugados nunca pasarán al semiplano complejo derecho.

Dado que se necesitan dos asíntotas, una para cada polo conjugado, que son los polos que no terminan en el cero sino en infinito, sus ángulos vienen dados por la siguiente ecuación

$$\theta = \frac{n 180^\circ}{z - p} \quad n = \pm 1, \pm 3, \dots \quad (114)$$

donde  $z$  y  $p$  siguen siendo la cantidad de ceros y polos en lazo abierto respectivamente; y  $n$  toma el valor sucesivamente de  $n = +1, -1, +3, -3, \dots$  hasta que se hayan encontrado los ángulos suficientes para cada asíntota.

Así quedan los ángulos necesarios para las dos asíntotas

$$\theta_1 = \frac{180^\circ}{1 - 3} = -90^\circ \quad (115)$$

$$\theta_2 = \frac{-180^\circ}{1 - 3} = +90^\circ \quad (116)$$

El punto de ruptura de salida sobre el eje real de estos polos que se vuelven conjugados para el sistema de la figura 48, se obtienen calculando

$\frac{dK}{ds} = 0$  para la ecuación característica la cual es

$$1 + K \frac{17.64 (s + 3)}{(s^3 + 9.026 s^2 - 35.37 s - 324.8)} = 0 \quad (117)$$

despejando  $K$

$$K = - \frac{(s^3 + 9.026 s^2 - 35.37 s - 324.8)}{17.64(s + 3)} \quad (118)$$

derivando respecto a  $s$

$$\frac{dK}{ds} = \frac{-35.28 s^3 - 317.97 s^2 - 955.31 s - 3857.69}{[17.64 (s + 3)]^2} = 0 \quad (119)$$

siendo las raíces

$$\begin{aligned} s_1 &= -7.35 \\ s_{2,3} &= -0.83 \pm j 3.76 \end{aligned} \quad (120)$$

hallando así el punto de ruptura  $-7.35$  en el eje real donde los polos son reales y dobles, y a partir del cual se vuelven complejos conjugados.

El tercer y último punto de importancia en el análisis del lugar geométrico de las raíces de la figura 48, corresponde al valor de  $K$  del punto de cruce del polo inestable al semiplano complejo izquierdo. A simple vista se observa que el punto de cruce es el origen pues el lugar de la raíz se encuentra sobre el eje real. Por lo tanto, solamente se necesitó encontrar el valor de  $K$  que hace esto posible. Esto se logró sustituyendo  $j\omega$  por  $s$  en la ecuación (118) y posteriormente haciendo  $\omega = 0$ . Así queda

$$K = \frac{324.8}{(17.64)(3)} \approx 6.14 \quad (121)$$

Por ende, la única condición necesaria para estabilizar al sistema del péndulo invertido con un controlador PI diseñado con un cero en  $-3$  introducido por la ganancia proporcional, es que  $K > 6.14$ .

Se escogió una ganancia  $K = 7$  y se ubicó el controlador en la trayectoria de realimentación quedando el sistema en lazo cerrado como en la figura siguiente

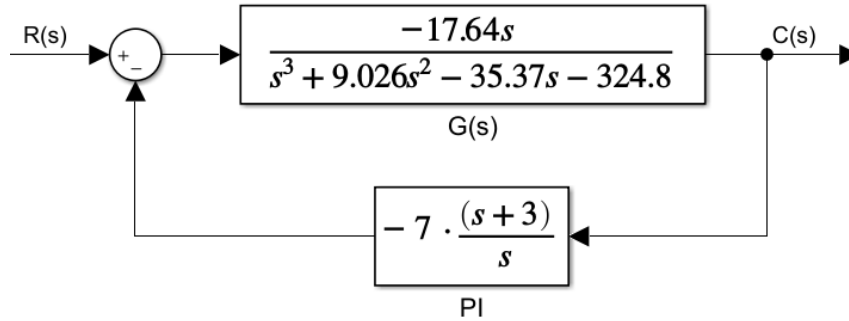


Figura 49. Controlador PI con  $b = 3$  y  $K = 7$  para el sistema en lazo cerrado del péndulo invertido sobre el carro.

Fuente: Autor.

Así la función de transferencia del controlador PI queda

$$-K \frac{(s + b)}{s} = -7 \frac{(s + 3)}{s} = -\left(7 + \frac{21}{s}\right) \quad (122)$$

donde la ganancia proporcional es 7 y la ganancia integral es 21.

Se hace hincapié que al ubicar el controlador en la trayectoria directa o en la trayectoria de realimentación, no altera el análisis del lugar geométrico de las raíces, lo único que cambia es la respuesta del sistema.

Ahora, su respuesta al impulso se muestra en la figura 50, la cual es estable. Al observar la respuesta del péndulo surgen varias preguntas importantes: ¿qué pasa con el carro?, ¿cuál es su respuesta al impulso? Bueno, para contestarlas se elaboró el diagrama de la figura 51.

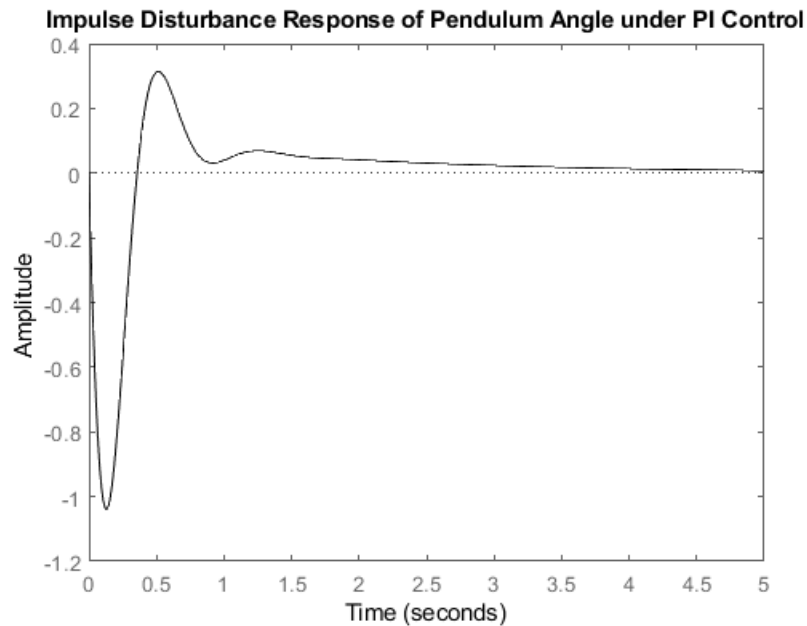


Figura 50. Respuesta al impulso del péndulo invertido con control PI.  
Fuente Autor.

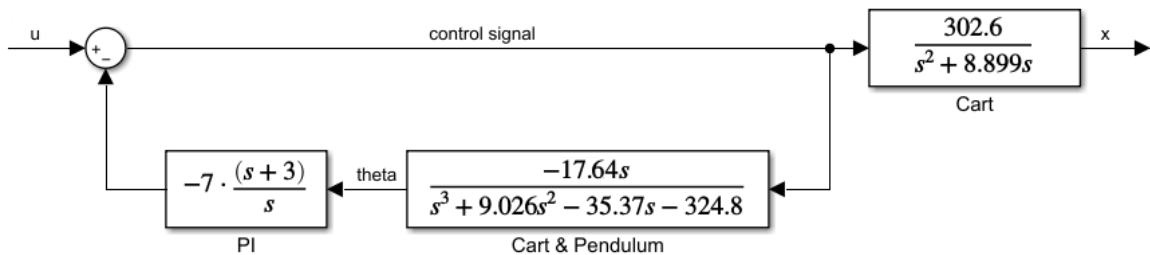


Figura 51. Diagrama de la respuesta del carro con control PI.  
Fuente: Autor.

La simulación de la respuesta al impulso del carro se presenta en la figura 52. Debido a que el controlador PI es solo para sistemas siso, es decir solo puede controlar un estado o variable a la vez, la posición del carro queda a la deriva y aunque no crece ilimitadamente debido a la respuesta al impulso, en el péndulo siempre habrá una ligera desviación de la vertical, por lo que el carro tenderá hacia los límites del riel sobre el cual se desplaza, cuyas medidas son

finitas. Por lo tanto, aunque el controlador PI estabiliza el ángulo del péndulo, este diseño no sería factible de implementar en el sistema físico real.

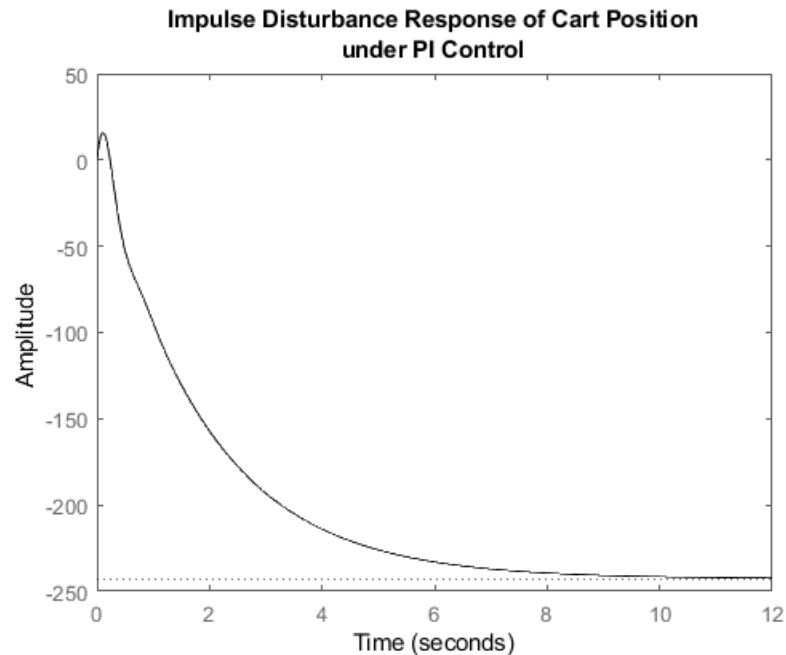


Figura 52. Respuesta al impulso del carro con control PI.

Fuente: Autor.

En el caso del PID, la ganancia derivativa le agregaría un cero al controlador PI. Así el sistema en lazo cerrado quedaría con dos ceros y tres polos. Dos polos irían a los dos ceros y el polo restante iría al cero en el infinito. Dependiendo de la ubicación de estos ceros agregados por las ganancias proporcional y derivativa, provocaría que los polos se muevan al semiplano complejo negativo.

Se eligió la ubicación de los ceros en  $-12$  y  $-15$  como se muestra en el lugar de las raíces de la figura 53. Aquí, de igual manera que en caso del control PI, son de importancia tres puntos. El primero, el punto de cruce del polo inestable al semiplano izquierdo. Segundo, los puntos de ruptura de salida y entrada de los dos polos que son conjugados para un intervalo de  $K$ . Tercero, el ángulo de la asíntota del polo que tiende al infinito, que en este caso se sabe que es  $180^\circ$  porque solamente es un polo.

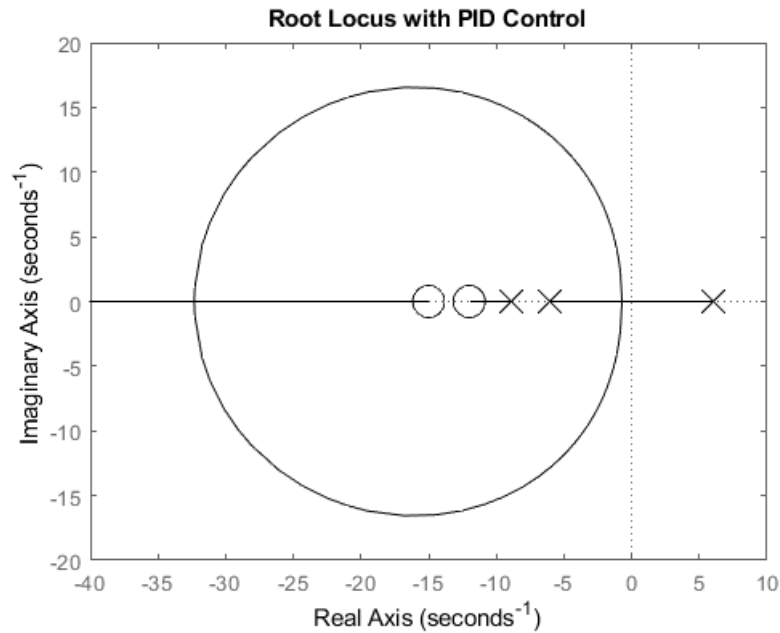


Figura 53. Lugar de las raíces con control PID.

Fuente: Autor.

El diagrama en lazo cerrado con control PID se muestra en la figura 54. Como se puede observar, en el análisis del lugar geométrico de las raíces no importa donde se ubique el controlador, ya sea en la trayectoria directa o de realimentación, ya que solamente es de importancia el polinomio característico de la función de transferencia en lazo cerrado, el cual permanece sin alteración.

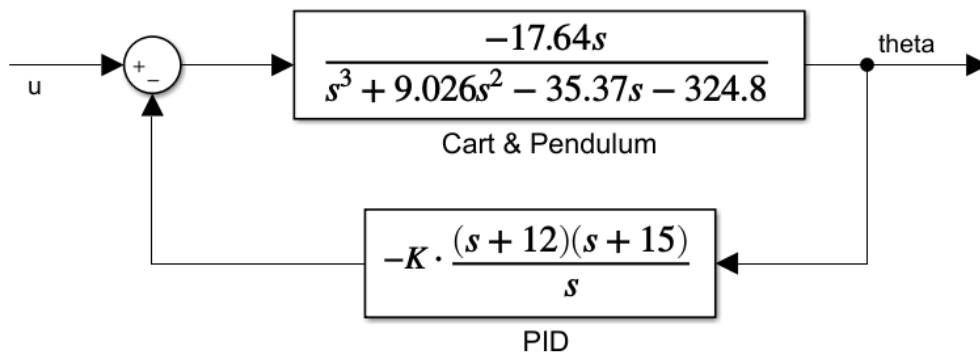


Figura 54. Diagrama en lazo cerrado con control PID.

Fuente: Autor.

El polinomio característico del sistema de la figura 54, con los polos y ceros cancelados, queda

$$1 + K \frac{17.64 (s + 12)(s + 15)}{s^3 + 9.026 s^2 - 35.37 s - 324.8} = 0 \quad (123)$$

El primer elemento de importancia en el lugar de las raíces, es la ganancia  $K$  que hace posible que el polo inestable pase al semiplano complejo izquierdo, por ello se sustituye  $j\omega$  en el polinomio característico (123) y se hace  $\omega = 0$ , siendo el punto de cruce el origen. Así el valor mínimo de  $K$  que estabiliza al sistema es

$$K = -\frac{-324.8}{17.64 (12)(15)} \approx 0.102 \quad (124)$$

El segundo elemento de importancia, son los puntos de ruptura de salida y entrada de los dos polos que son complejos conjugados en un intervalo de  $K$ . Para ello, se deriva  $K$  respecto a  $s$  en el polinomio característico (123).

$$\frac{dK}{ds} = \frac{-17.64 s^4 - 952.56 s^3 - 14448.43 s^2 - 68777.65 s - 42388.92}{[17.65 (s + 12)(s + 15)]^2} = 0 \quad (125)$$

Las raíces del numerador de la ecuación (125)

$$\begin{aligned} s_1 &= -32.317 \\ s_2 &= -13.051 \\ s_3 &= -7.91 \\ s_4 &= -0.72 \end{aligned} \quad (126)$$

Como las raíces  $s_2$  y  $s_3$  no pertenecen al lugar geométrico de las raíces en el eje real, solo nos quedan las raíces  $s_1$  y  $s_4$ , que son el punto de ruptura de entrada y salida respectivamente. Para encontrar los valores de  $K$  de estos puntos,

solamente se sustituyen en el polinomio característico (123) dando como resultado

$$\begin{aligned} K_{s1} &= 3.79 \\ K_{s4} &= 0.104 \end{aligned} \quad (127)$$

Por ende los polos en lazo abierto más a la derecha del lugar geométrico de las raíces son complejos conjugados en el intervalo de  $3.79 > K > 0.104$ .

Se eligió para el controlador PID un valor de  $K = 4$ , que es mayor al punto de ruptura de entrada de los polos complejos conjugados  $K_{s1}$ . Simulando su respuesta al impulso se obtuvo la figura 55.

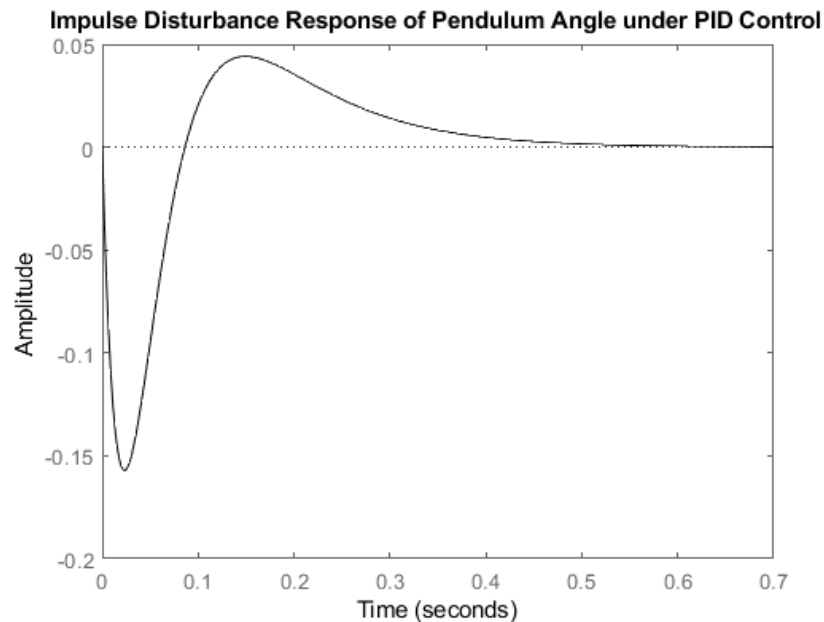


Figura 55. Respuesta al impulso del péndulo invertido con control PID.

Fuente: Autor.

Se observa que al aplicar un impulso al péndulo, éste regresa a reposo. Nuevamente, ¿qué pasa con el carro? Para responder la pregunta se presenta el diagrama de la figura 56 y su respuesta al impulso en la figura 57.



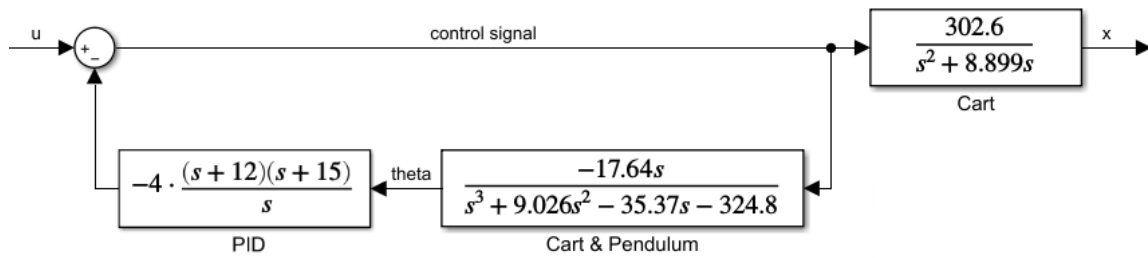


Figura 56. Diagrama para la obtención de la respuesta del carro con control PID.  
Fuente: Autor.

En la figura 57, la posición del carro al igual que en el controlador PI, no crece ilimitadamente debido a la excitación impulso, pero tampoco regresa a su estado inicial cero. Por lo que es necesario hacer la misma observación, y es que en el péndulo siempre habrá una ligera desviación de la vertical, y al no haber control sobre el carro, éste tiende hacia los límites del riel. Por lo tanto, aunque el controlador PID también estabiliza el ángulo del péndulo, este diseño no sería factible de implementar en el sistema físico real.

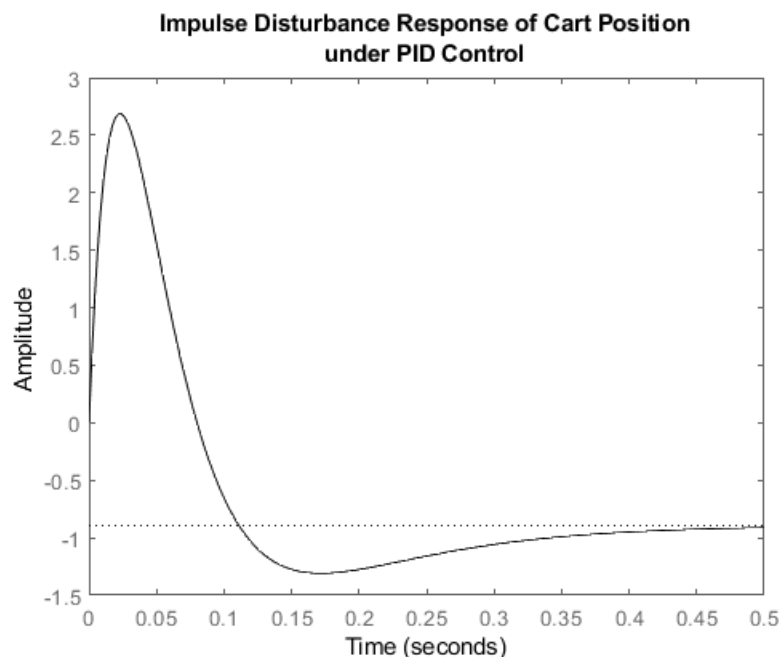


Figura 57. Respuesta al impulso del carro con control PID.  
Fuentes: Autor.

## 2.8 DISEÑO DEL CONTROLADOR LQR PARA EL PÉNDULO INVERTIDO SOBRE EL CARRO

La representación en espacio de estados del sistema del péndulo invertido en tiempo continuo está dada por las ecuaciones (107) y (108). Antes de empezar el diseño del controlador LQR, primero se asegura que el sistema sea completamente controlable. Esto quiere decir, que si es completamente controlable, se puede ubicar el estado del sistema en cualquier lugar que se desee en un tiempo finito (bajo restricciones físicas del sistema). Para que sea completamente controlable, la matriz de controlabilidad de la ecuación (63) debe tener rango  $n$ , donde el rango de una matriz es el número de filas (o columnas) linealmente independientes, y donde  $n$  es el número de variables de estado del sistema.

Para todo este acápite 2.8 se utilizó el script de Matlab *Inverted\_Pendulum\_LQR\_design.m*<sup>11</sup>. Se encontró que el rango de la matriz de controlabilidad del sistema es 4. Dado que el rango es igual a la cantidad de variables de estado, el sistema es completamente controlable. El sistema de control en lazo cerrado se muestra a continuación

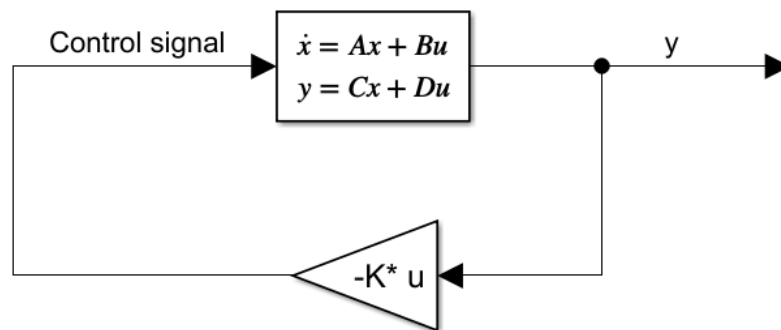


Figura 58. Control LQR para el sistema del péndulo invertido sobre el carro.

Fuente: Autor.

Para el diseño del controlador LQR se asume que se pueden medir todas las variables de estados. Posteriormente, se escogieron valores fáciles para las matrices  $Q$  y  $R$

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (128)$$

$$R = 1 \quad (129)$$

El comando `lqr` en Matlab resuelve la ecuación de Ricatti, con el que se obtuvo la matriz de ganancia  $K$  siguiente

$$K = [-1.0000 \quad -1.3678 \quad -249.6242 \quad -40.8937] \quad (130)$$

Al simular la respuesta al impulso del sistema en lazo cerrado con la matriz de realimentación  $K$  se obtuvo la figura 59, la cual muestra los estados de interés  $x$  y  $\theta$ , el desplazamiento angular del péndulo y la posición lineal del carro respectivamente. Como se puede observar el sistema es estable, y en contraste con el control PID, en el control LQR se tiene control de todos los estados, retornando la posición del carro a su estado inicial cero ante la perturbación impulso. Por lo tanto, este diseño sería factible de implementar en el sistema físico real.

La estabilidad se confirma mediante la posición de los polos de lazo cerrado los cuales se encuentran en el semiplano complejo izquierdo, por lo que sus partes reales son negativas y se presentan a continuación

$$\begin{aligned} s_1 &= -303.25 \\ s_{2,3} &= -6.03 \pm j 0.16 \\ s_4 &= -1 \end{aligned} \quad (131)$$

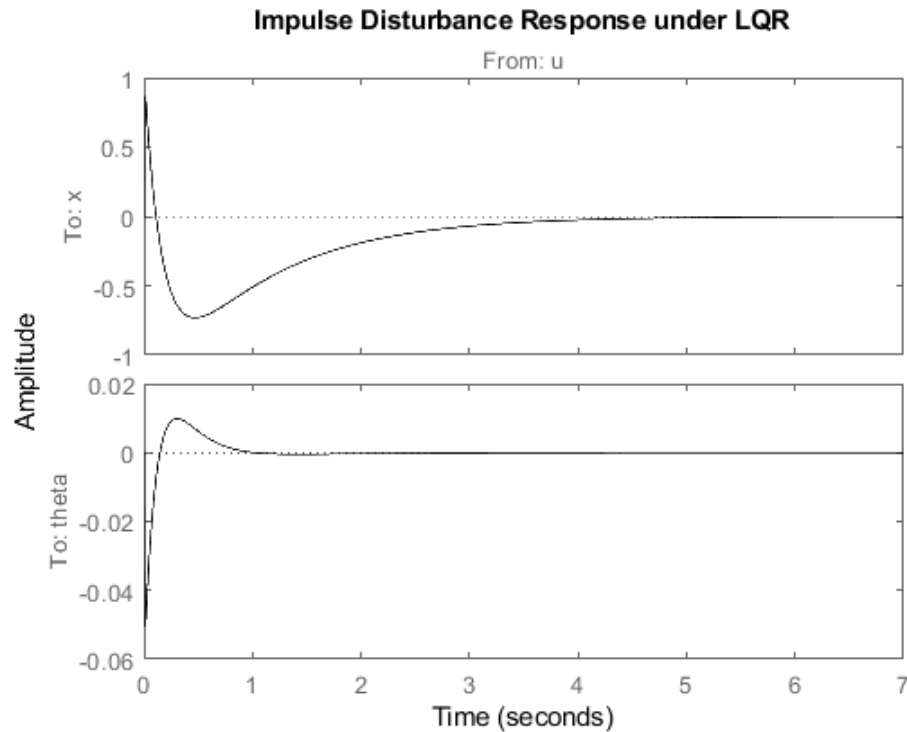


Figura 59. Respuesta al impulso de la posición del carro  $x$  y del péndulo  $\theta$  bajo control LQR.

Fuente: Autor.

## 2.9 IMPLEMENTACIÓN DE LOS CONTROLADORES PID Y LQR EN EL SISTEMA FÍSICO REAL

Para rebasar la brecha del diseño a la implementación, se debe estar consciente de las limitaciones que tiene el modelo del sistema de control. Y es que en el diseño no se tomó en cuenta las no linealidades que todos los sistemas físicos poseen. Por ejemplo, no se tomó en cuenta la saturación tanto del controlador como de la planta, que se traduce a que el actuador puede entregar energía infinita y la planta puede, de igual manera, procesar energía infinita, lo que es irreal. El hecho de que todos los sistemas sean no lineales, no debería de ser un desalentador para diseñar e implementar los controladores lineales, puesto que todo sistema tiene regiones cuyo comportamiento es lineal. Si se desea obtener una respuesta de un modelo más apegado a la realidad, se tendrían que modelar las no linealidades faltantes.

Dado que los algoritmos de control PID y LQR diseñados se implementan en la tarjeta microcontroladora Arduino Mega 2560, es necesario pasar del tiempo continuo al tiempo discreto. Siendo la acción de control PID analógico de la ecuación (33) que se muestra nuevamente a continuación

$$m(t) = K \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d(\tau) + T_d \frac{de(t)}{dt} \right) \quad (132)$$

Si se deja la ecuación (132) en función de las ganancias en lugar de los tiempos integral y derivativo, queda

$$m(t) = K_P e(t) + K_I \int_0^t e(\tau) d(\tau) + K_D \frac{de(t)}{dt} \quad (133)$$

La función de transferencia pulso para el controlador PID digital es

$$G_D(z) = \frac{M(z)}{E(z)} = K_{Pd} + \frac{K_{Id}}{1 - z^{-1}} + K_{Dd}(1 - z^{-1}) \quad (134)$$

o bien

$$G_D(z) = \frac{M(z)}{E(z)} = \frac{(K_{Pd} + K_{Id} + K_{Dd}) - (K_{Pd} + 2K_{Dd})z^{-1} + K_{Dd}z^{-2}}{1 - z^{-1}} \quad (135)$$

donde

$$\begin{aligned} K_{Pd} &= K - \frac{KT}{2T_i} = K - \frac{K_{Id}}{2} \\ K_{Id} &= \frac{KT}{T_i} \\ K_{Dd} &= \frac{KT_d}{T} \end{aligned} \quad (136)$$

en función de la ecuación (132), o bien, en función de la ecuación (133)

$$\begin{aligned} K_{Pd} &= K_P - \frac{K_I T}{2} = K_P - \frac{K_{Id}}{2} \\ K_{Id} &= K_I T \\ K_{Dd} &= \frac{K_D}{T} \end{aligned} \quad (137)$$

Basados en el controlador PID de la figura 56 se tiene

$$G_D(s) = - \left[ 4 \frac{(s + 12)(s + 15)}{s} \right] \quad (138)$$

o bien

$$G_D(s) = - \left[ 108 + \frac{720}{s} + 4s \right] \quad (139)$$

Al pasar las ganancias para el control digital utilizando la ecuación (137) queda

$$\begin{aligned} K_{Pd} &= K_P - \frac{K_I T}{2} = 108 - \frac{720 T}{2} \\ K_{Id} &= K_I T = 720 T \\ K_{Dd} &= \frac{4}{T} \end{aligned} \quad (140)$$

Ahora, las ganancias de la ecuación (140) varían en dependencia del tiempo de muestreo  $T$ . Si se elige un tiempo de muestreo  $T = 0.005 \text{ seconds}$  queda

$$\begin{aligned} K_{Pd} &= 108 - \frac{(720)(0.005)}{2} = 106.2 \\ K_{Id} &= (720)(0.005) = 3.6 \\ K_{Dd} &= \frac{4}{0.005} = 800 \end{aligned} \quad (141)$$

Así al implementar el controlador PID analógico de la ecuación (139) en su forma digital (135) y sustituyendo las ganancias (141) queda

$$G_D(z) = - \left[ \frac{(106.2 + 3.6 + 800) - (106.2 + 2 * 800)z^{-1} + 800 z^{-2}}{1 - z^{-1}} \right] \quad (142)$$

ó

$$G_D(z) = - \left[ \frac{909.8 - 1706.2 z^{-1} + 800 z^{-2}}{1 - z^{-1}} \right] \quad (143)$$

Si se introduce el signo negativo

$$G_D(z) = \frac{-909.8 + 1706.2 z^{-1} - 800 z^{-2}}{1 - z^{-1}} \quad (144)$$

La implementación en diagrama de bloques de la ecuación (135) es

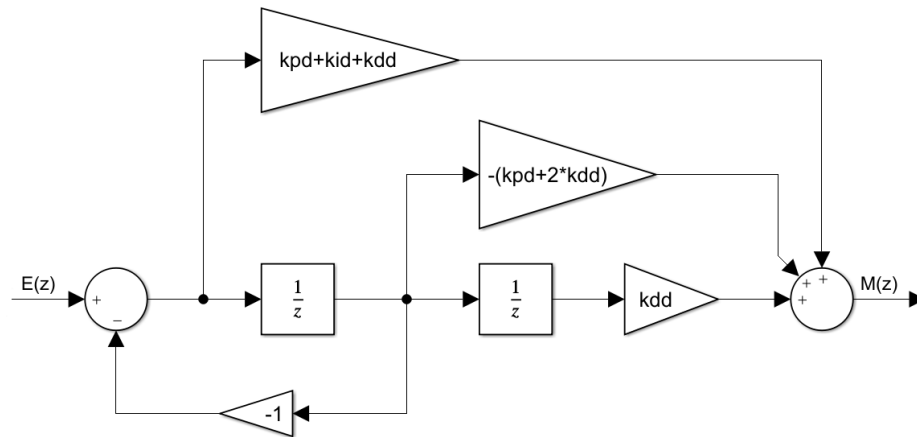


Figura 60. Diagrama de bloques de la realización de un controlador PID digital.

Fuente: Autor.

Sustituyendo los coeficientes de la ecuación (144)

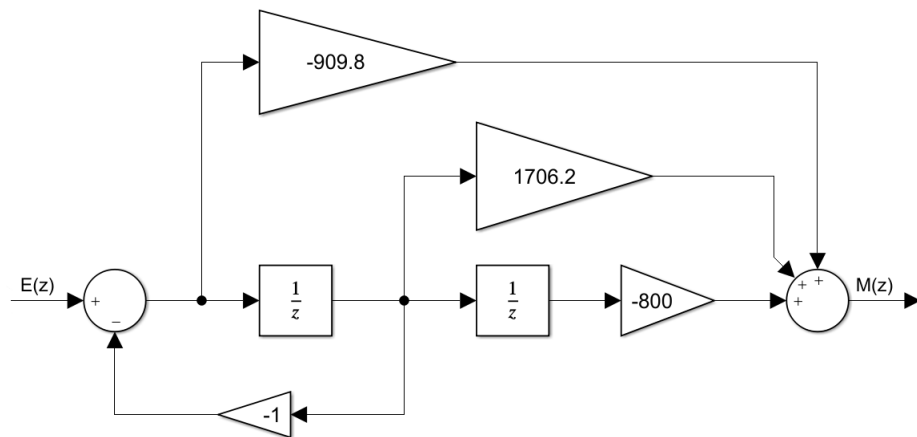


Figura 61. Diagrama de bloques del controlador PID digital con las ganancias para un tiempo de muestreo de 5 milisegundos.

Fuente: Autor.

El diagrama en lazo cerrado para el péndulo invertido en tiempo discreto con control PID digital y tiempo de muestreo de 5 milisegundos queda

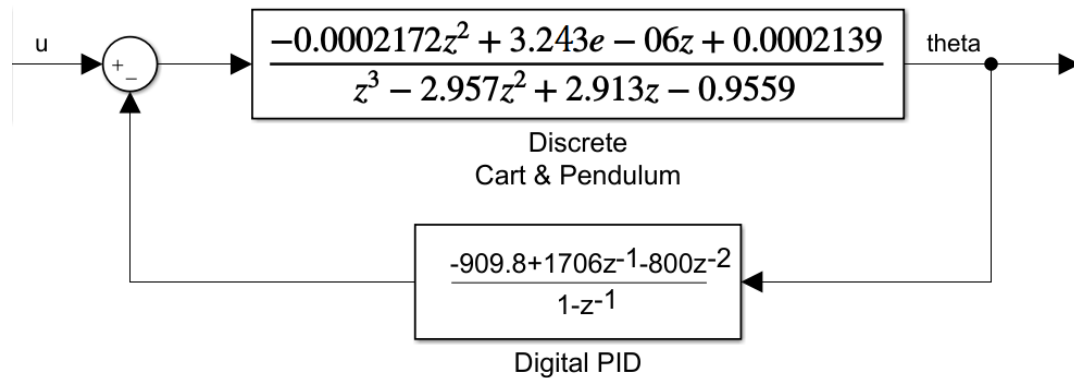


Figura 62. Diagrama en lazo cerrado del péndulo invertido en tiempo discreto con control PID digital y tiempo de muestreo de 5 milisegundos.

Fuente: Autor.

La simulación de la respuesta al impulso del péndulo es

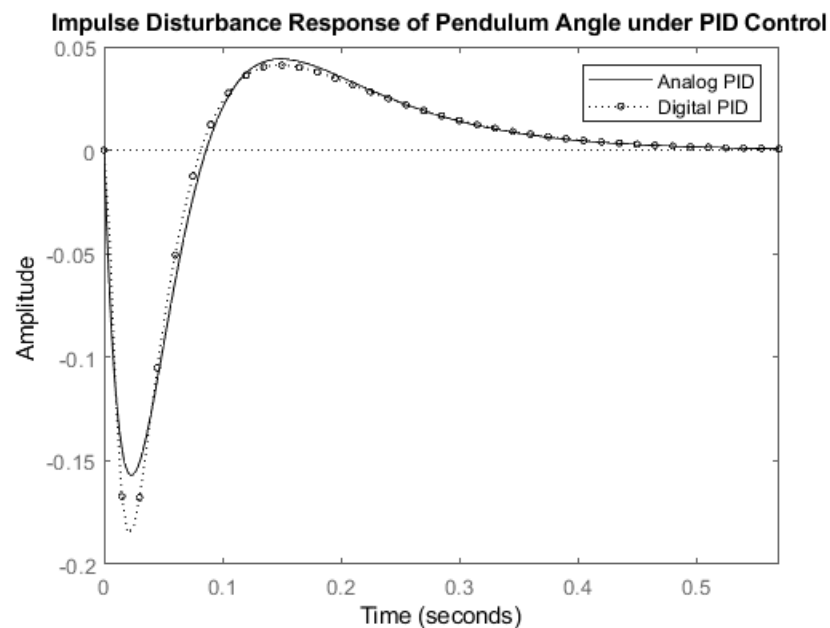


Figura 63. Respuesta al impulso del péndulo invertido con control PID digital y analógico.

Fuente: Autor.



Para conocer la respuesta del carro se tiene el diagrama de la figura 64 con las funciones de transferencias discretizadas

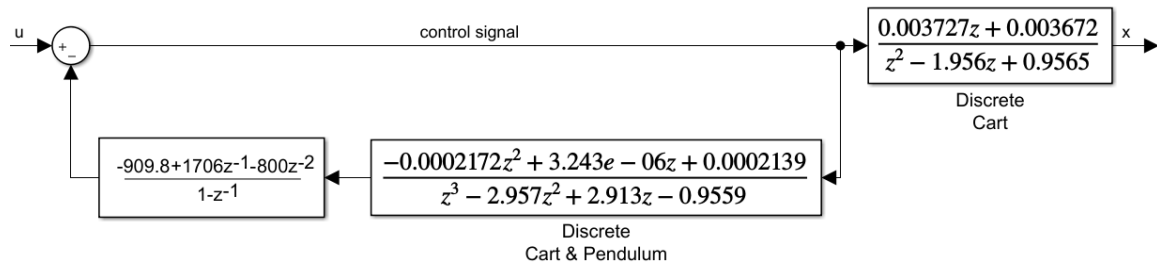


Figura 64. Diagrama de bloques para la obtención de la respuesta del carro con controlador PID digital.

Fuente: Autor.

Y su respuesta al impulso en comparación con el PID analógico se muestra a continuación

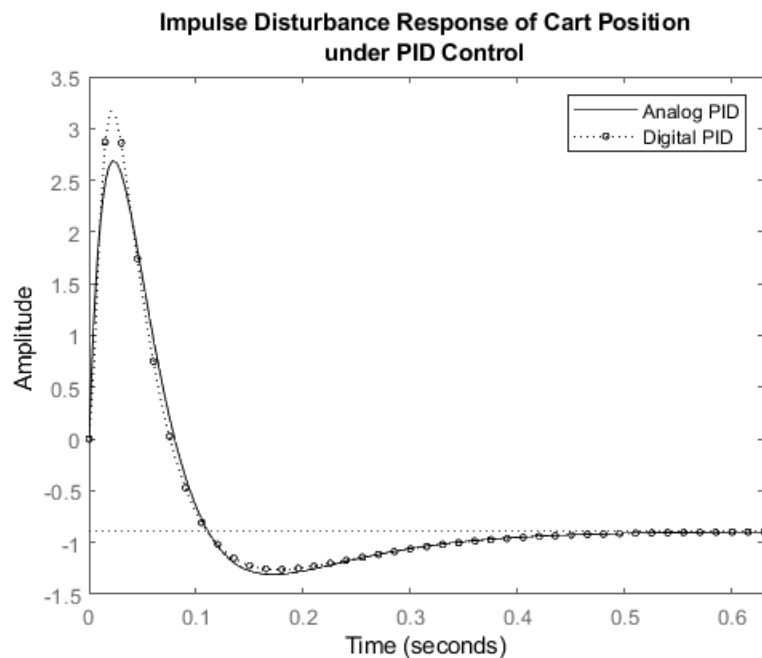


Figura 65. Respuesta al impulso del carro con control PID digital y analógico.

Fuente: Autor.

Con el programa en Simulink *Inverted\_Pendulum\_PID\_implementation\_embedded.slx* que corre en el Arduino Mega 2560 y el programa *Inverted\_Pendulum\_PID\_implementation\_embedded.slx* que corre como host en el computador, se logró implementar el controlador PID y obtener las gráficas del sistema físico real. En la figura 66 se expone el ángulo del péndulo, el cual comienza en su posición vertical hacia abajo  $180^\circ$  y en un tiempo aproximado de 4 segundos se cambia manualmente la varilla a su posición vertical hacia arriba donde el control PID empieza a actuar manteniendo el péndulo vertical a  $0^\circ$ .

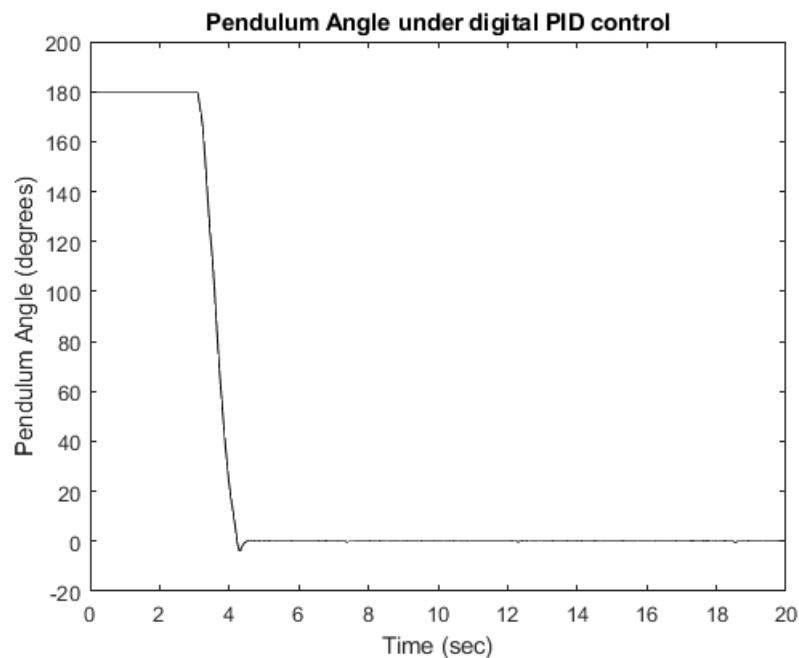


Figura 66. Ángulo del péndulo bajo control PID digital en el sistema físico real.

Fuente: Autor.

Ahora, en la figura 67 se muestra la posición del carro, sobre la cual no hay control alguno y se comporta errático desplazándose hacia los límites del riel, por lo que con las manos se le aplicó perturbaciones a la varilla con el objetivo de mantener el carro dentro de los límites.

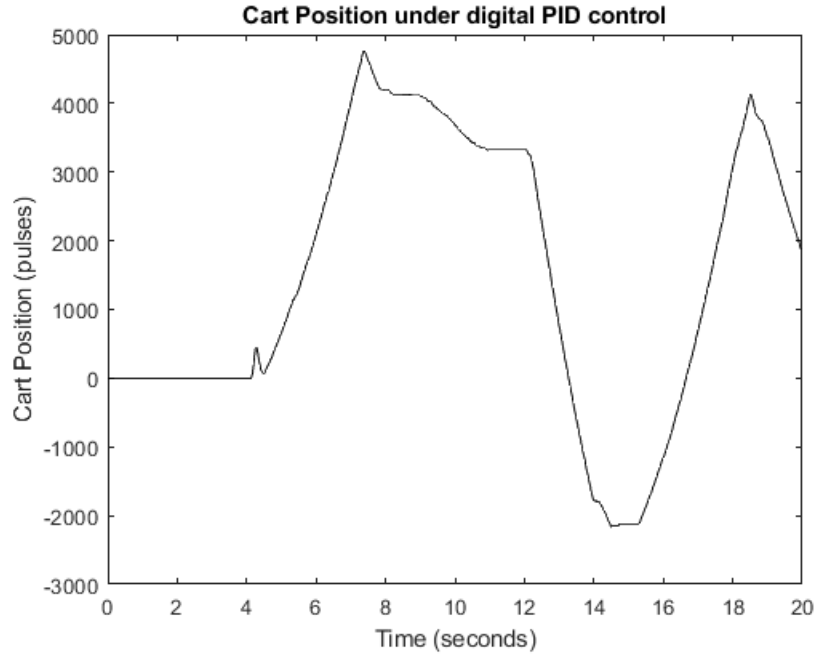


Figura 67. Posición del carro bajo control PID digital en el sistema físico real.

Fuente: Autor.

En el control LQR, de igual manera que con el PID, es necesario pasar al tiempo discreto para la implementación del algoritmo de control en la tarjeta microcontroladora Arduino Mega 2560. Así el modelo discreto del péndulo invertido sobre el carro de las ecuaciones (107) y (108) con un tiempo de muestreo de 5 milisegundos es

$$\begin{bmatrix} \dot{x}(k+1) \\ \ddot{x}(k+1) \\ \dot{\theta}(k+1) \\ \ddot{\theta}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.00489 & 0 & 0 \\ 0 & 0.9565 & 0 & 0 \\ 0 & 6.38 \times 10^{-6} & 1 & 0.004999 \\ 0 & 0.002536 & 0.1825 & 0.9998 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \\ \theta(k) \\ \dot{\theta}(k) \end{bmatrix} + \begin{bmatrix} 0.003727 \\ 1.48 \\ -0.0002172 \\ -0.08623 \end{bmatrix} u(k) \quad (145)$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \\ \theta(k) \\ \dot{\theta}(k) \end{bmatrix} \quad (146)$$

Con una modificación en las matrices Q y R, dándole más peso a la posición del carro y al ángulo del péndulo, se busca disminuir el error en estas variables de

estado. También se le dio más peso a la matriz R, lo que relaja la señal de control, evitando así la saturación del actuador.

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (147)$$

$$R = 1 \times 10^4 \quad (148)$$

Con el comando `dlqr` en Matlab que devuelve la matriz de ganancia de realimentación óptima  $K_d$  para el sistema del péndulo invertido en tiempo discreto

$$K_d = [-0.0963 \quad -0.1007 \quad -15.7324 \quad -2.5167] \quad (149)$$

Se simula su respuesta al impulso y se obtiene la figura 68. Se asume en esta simulación que se realimentan todos los estados y que no hay saturación en el actuador.

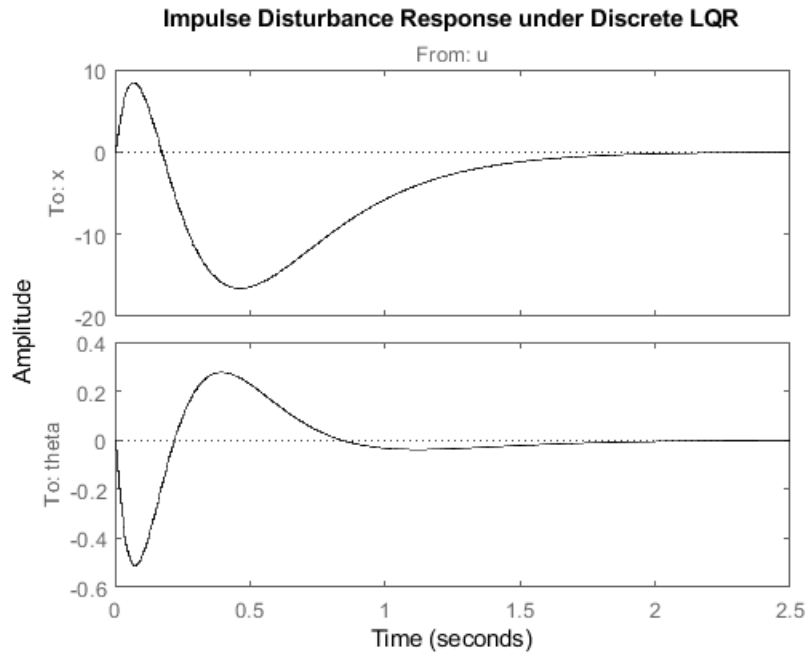


Figura 68. Respuesta al impulso de la posición del carro  $x$  y del péndulo  $\theta$  bajo control LQR discreto.

Fuente: Autor.

Utilizando los programas de Simulink *Inverted\_Pendulum\_LQR\_implementation\_embedded.slx* que corre en el Arduino Mega 2560 y el programa *Inverted\_Pendulum\_LQR\_implementation\_host.slx* que corre como host en el computador se logró implementar el controlador LQR discreto y obtener las siguientes gráficas del sistema físico real. La figura 69 muestra el ángulo del péndulo en grados, el cual comienza en su posición vertical hacia abajo en  $180^\circ$  y a los tres segundos aproximadamente, se ubica manualmente a  $0^\circ$  donde empieza a actuar el controlador LQR. Se puede ver que el regulador mantiene a la varilla estable en su posición vertical  $0^\circ$ .

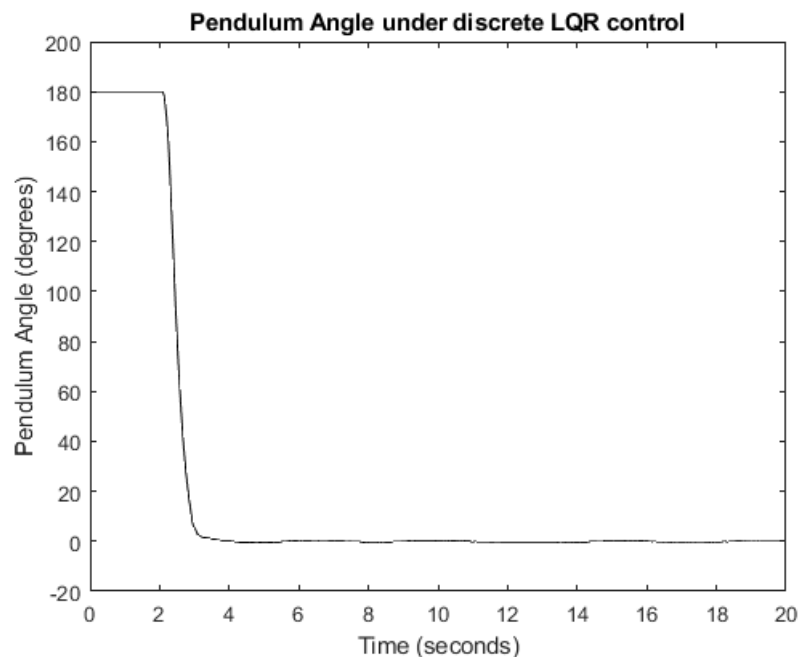


Figura 69. Ángulo del péndulo bajo control LQR discreto en el sistema físico real.

Fuente: Autor.

En el caso del carro se tiene la figura 70, el cual se mantiene en un rango de 1000 pulsos del encoder aproximadamente. No fue necesario estarle aplicando manualmente perturbaciones para mantenerlo dentro del riel de desplazamiento.

Para poder implementar el controlador LQR en el sistema físico real, fue necesario realimentar todas las variables de estado, las cuales son las

posiciones y velocidades del péndulo y del carro. Pero, como suele suceder en la realidad y aquí no es la excepción, no es posible medir todos los estados.

En este caso solamente se están midiendo las posiciones del carro y del péndulo, y las velocidades se obtuvieron derivando las posiciones, lo cual no es aconsejable; ya que la derivación de una señal siempre disminuye la relación de señal a ruido porque el ruido generalmente fluctúa más rápidamente que la señal. Por lo tanto, es aconsejable medir directamente los estados o estimar aquellos que no pueden ser medidos, pero esto es para aquél que desee ahondar más en este proyecto.

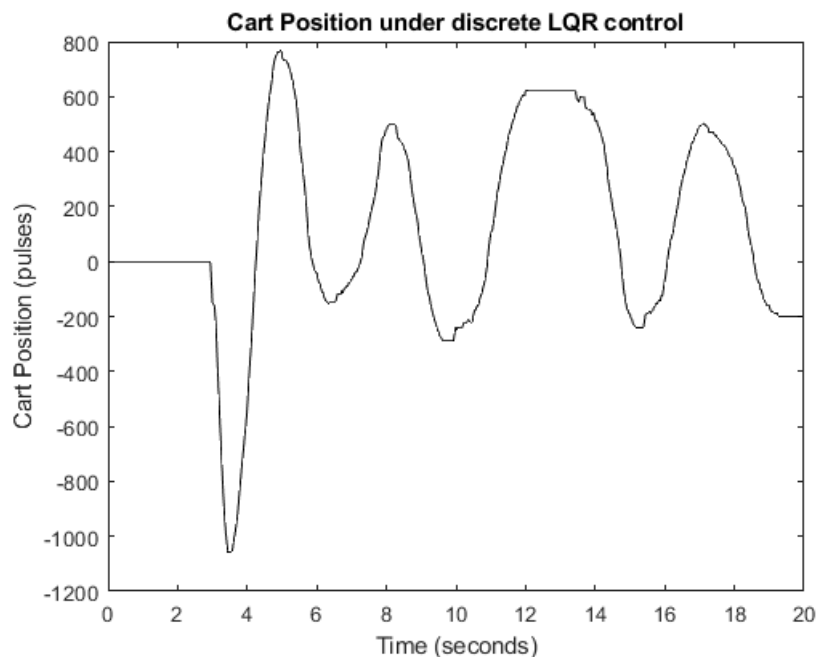


Figura 70. Posición del carro bajo control LQR discreto en el sistema físico real.

Fuente: Autor.

Por esta razón, la posición del carro de la figura 70 no es igual a la simulación del modelo en la figura 68. Primero, el impulso es una idealización. Segundo, todos los modelos son erróneos. Esto quiere decir que el modelo utilizado para el péndulo invertido es solo una aproximación de la realidad, que en este caso logra su cometido que es en conjunto con el control LQR, estabilizar el péndulo en su posición vertical hacia arriba. Agregando, existen no

linealidades que no se incluyeron en el modelo como son la zona muerta y la saturación del motor, que afectan el desempeño del sistema de control. Por lo tanto, debido a la incertidumbre del modelo, se debe utilizar la integración de igual manera que se utiliza en el control clásico, con el objeto de eliminar el error en los estados.

En las figuras 71 y 72 se aprecian las velocidades del péndulo y del carro que confirman la presencia del ruido debido a la derivación de los estados medidos directamente.

Haciendo un análisis comparativo del desempeño en la ejecución de los compensadores PID y LQR, se observó que el controlador LQR es más adecuado para el péndulo invertido (sistema multivariable), ya que permite el control de todos los estados. Esto favorece a la ejecución del LQR porque al tener control sobre la posición del carro, éste no busca los límites del riel donde el ángulo del péndulo se vuelve inestable. No así en el caso del PID, donde solamente se pudo controlar el ángulo del péndulo y no la posición del carro, quedando éste a la deriva. Esta es una limitante del controlador PID, resultando adecuado para sistemas siso donde se controla una sola variable por lazo cerrado.

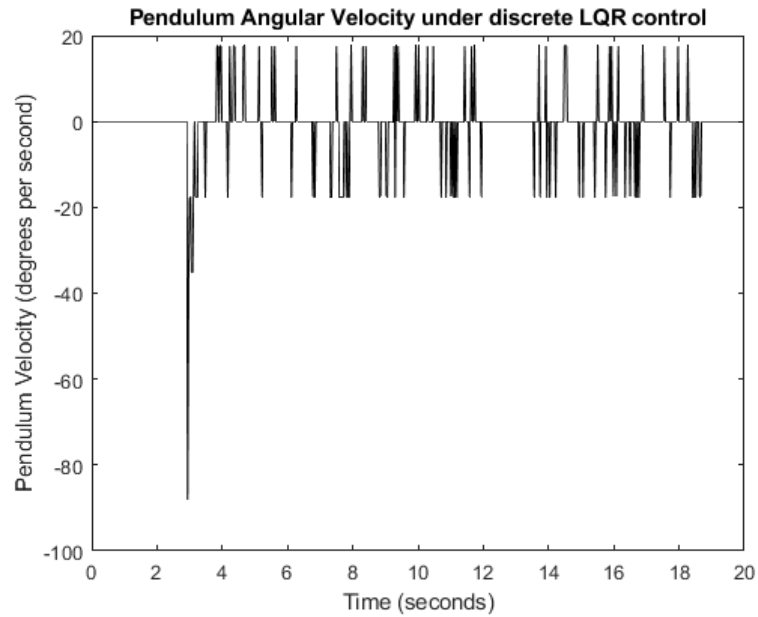


Figura 71. Velocidad angular del péndulo en el sistema físico real bajo control LQR discreto.

Fuente: Autor.

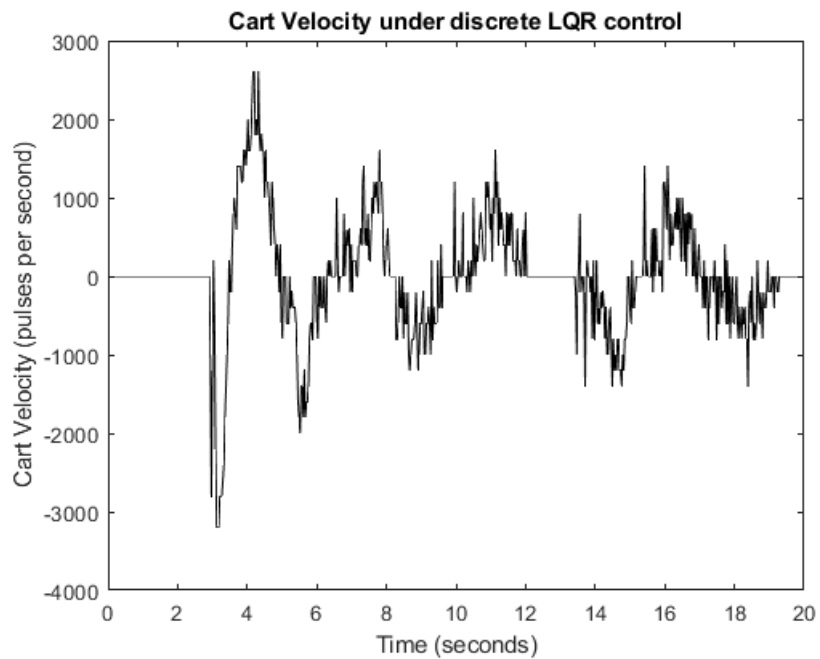


Figura 72. Velocidad del carro en el sistema físico real bajo control LQR discreto.

Fuente: Autor.



## CONCLUSIONES

Se construyó el prototipo para el péndulo invertido sobre el carro utilizando la estructura mecánica de una impresora de papel. Ésta facilitó todo el proceso de construcción, ya que proporcionó la mayoría de partes mecánicas para el funcionamiento completo, permitiendo la aplicación de la teoría de control utilizando el software de Matlab, logrando el cumplimiento de los objetivos subsiguientes.

Se elaboró un modelo del sistema basado en principios físicos que brindan la estructura de las ecuaciones matemáticas, las cuales junto con la técnica de los mínimos cuadrados para la identificación del carro y el estudio de las oscilaciones del péndulo, consolidan el modelo matemático final sobre el cual se basó el diseño de los controladores con el objeto de estabilizar el péndulo invertido de naturaleza inestable.

Se diseñó el controlador PID, utilizando la técnica del diagrama de las raíces, logrando estabilizar el ángulo del péndulo invertido, no así la posición del carro.

Se diseñó el controlador LQR utilizando la técnica de espacio de estados, que brindó las ganancias óptimas de realimentación de las variables de estado estabilizando el ángulo del péndulo invertido y manteniendo la posición del carro dentro de los límites del riel.

Se implementaron los controladores digitales (PID y LQR) en el prototipo del péndulo invertido y se compararon sus desempeños concordando con lo expuesto en el diseño.

Se elaboró un manual de usuario que logra orientar al usuario acerca del diseño y funcionamiento del prototipo del péndulo invertido.

## RECOMENDACIONES

Se sugiere la utilización de un motor con mayor velocidad y voltaje de operación para evitar que la señal de control se sature a un menor valor dotando de una mayor robustez al sistema. Se recomienda la construcción de una tarjeta de circuito impreso PCB que dé mayor seguridad a las conexiones de los componentes electrónicos.

En el control PID, se sugiere agregar otro lazo que controle la posición del carro y permita su permanencia dentro de los límites del riel.

En el control LQR, se recomienda diseñar un observador para estimar los estados no medidos directamente como son las velocidades del carro y del péndulo para mejorar el desempeño en la ejecución del compensador. También se recomienda la inclusión de un integrador para eliminar el error en estado estable de la posición del carro.

Se propone la elaboración de una serie de guías de laboratorio donde se desarrollen los aspectos teóricos y prácticos expuestos en este trabajo.

## BIBLIOGRAFÍA

- Banzi, M. (2011). *Getting Started with Arduino*. O'Reilly.
- Calvo Rolle, J. (2010). *Scilab: Programación y Simulación*. Ra-Ma S.A. Editorial y Publicaciones.
- Cannon, R. H. (1967). *Dynamics of Physical Systems*. McGraw-Hill.
- Duarte V., Ó. (s.f.). *Análisis de sistemas dinámicos lineales*. Bogotá: Universidad Nacional de Colombia.
- Gerez Greisser, V., & Murray-Lasso, M. (1972). *Teoría de Sistemas y Circuitos*. Representaciones y Servicios de Ingeniería S. A.
- J. Astrom, K., & Hagglund, T. (2009). *Control PID avanzado*. Madrid: Pearson Educación S. A.
- Kuo, B. C. (1996). *Sistemas de Control Automático*. Prentice-Hall Hispanoamérica S.A.
- Lago, G., & Benningfield, L. M. (1984). *Teoría de Sistemas y Circuitos*. Editorial Limusa.
- Lynch, W. A., & Truxal, J. G. (1961). *Introductory System Analysis: Signals and Systems in Electrical Engineering*. York, PA.: McGraw-Hill.
- Matlab. Descripción general*. (s.f.). Obtenido de MathWorks: <https://la.mathworks.com/products/matlab.html>
- Metzler, P. (s.f.). *Modelling and Simulation using Matlab*.
- Nise, N. S. (2009). *Sistemas de Control para Ingeniería*. Grupo Editorial Patria.
- Ogata, K. (1980). *Ingeniería de Control Moderna*. Prentice-Hall Hispanoamérica.
- Ogata, K. (1987). *Dinámica de Sistemas*. Prentice-Hall Hispanoamérica S.A.
- Ogata, K. (1996). *Sistemas de Control en Tiempo Discreto*. Prentice Hall Hispanoamérica.
- Phelan, R. M. (1977). *Automatic Control Systems*. Cornell University Press.
- Puente h para control de motores*. (s.f.). Obtenido de Ingeniería mecafenix: <http://www.ingmecafenix.com/electronica/puente-h-control-motores/>
- Simulink. Descripción general*. (s.f.). Obtenido de Mathworks: <https://es.mathworks.com/products/simulink.html>

Tsipkin, A. G. (1985). *Manual de Matemáticas para la Enseñanza Media*. Editorial Mir Moscú.

Venturin, M., & Poles, S. (s.f.). *Controller for an Inverted Pendulum*. Obtenido de Openeering:

[http://www.openeering.com/sites/default/files/Inverted\\_Pendulum.pdf](http://www.openeering.com/sites/default/files/Inverted_Pendulum.pdf)

## **ANEXOS**

### **Anexo 1. Entrevistas**

#### **Entrevista al MBA Ing. Francisco Galo, Jefe de Dpto. de Electrónica**

- ¿Qué opinión tiene sobre el Laboratorio de Automatización y las prácticas que realizan allí los estudiantes?

Opino que en la Universidad hace falta una constante actualización en cuanto a equipos se refiere. Pues la ingeniería es una ciencia que día a día avanza de manera exponencial, y de cierta manera nos hemos ido quedando con equipos y prácticas que ya van pasando a la historia. Los laboratorios deberían de cambiar periódicamente, pero realmente es difícil de conseguir ya que esto implica un gasto económico, un gasto que muchas veces la Universidad no está en condiciones de pagar.

- Luego de explicarle sobre el péndulo invertido. ¿Considera útil la propuesta de diseñar un péndulo invertido para que sea instalado en el laboratorio y tanto alumnos como estudiantes puedan valerse de esta herramienta?

Me parece una propuesta muy interesante y por supuesto de gran utilidad para la universidad, ya que a como decía en la anterior pregunta, equiparse con nuevas herramientas es costoso y resulta muy provechoso que los mismos estudiantes nos provean equipos de gran utilidad para las generaciones nuevas. Y no sólo hablando de la parte económica, sino también de la parte didáctica. Hacerse con ese péndulo viene a facilitar la enseñanza y a renovar un poco las actuales clases prácticas que se realizan en los laboratorios.

## **Entrevista al Ing. Bayardo Juárez, Responsable del Laboratorio de Automatización**

- ¿Considera que el Laboratorio de Automatización ya cuenta con todas las herramientas necesarias para las prácticas de sus estudiantes?  
Pienso que le hace falta un poco.
- ¿Existen prácticas de laboratorio sobre modelado de sistemas?  
No hay.
- ¿Sabe si hay prácticas sobre identificación de parámetros de un sistema y cómo ha observado el desarrollo de los estudiantes al momento de realizar estas prácticas?  
No hay.
- ¿Cuenta el laboratorio de Automatización con una herramienta para realizar prácticas sobre compensadores PID y LQR?  
En la clase no hay laboratorios sobre compensadores PID y LQR.
- ¿De qué manera cree usted que las prácticas de laboratorio podrían fortalecerse?  
Recurriendo a nuevas herramientas y nuevos equipos para el laboratorio de Automatización.
- ¿Sabe qué es un péndulo invertido?  
Lo desconozco.
- Luego de explicarle un poco sobre el péndulo ¿considera que un péndulo invertido sería una herramienta que ayudaría tanto a alumnos como a maestros a desarrollar los contenidos de Sistemas de Control?

Ahora que me explicaron un poco me llama mucho la atención y me gustaría saber un poco más de él por medio de distintas prácticas, considero que sería de gran utilidad.

- ¿Consideraría importante agregar algunas prácticas de laboratorio haciendo uso de una herramienta como el péndulo invertido?

Por supuesto, es una propuesta muy interesante y nos beneficiaría de gran medida.

- ¿Qué opina de que el laboratorio cuente con herramientas diseñadas por los mismos estudiantes con el propósito de fortalecer los conocimientos adquiridos en clases?

Considero muy importante señalar que me parece de gran utilidad que los mismos estudiantes aporten sus ideas y sus proyectos para que sean utilizados en el proceso de enseñanza de los nuevos alumnos, además que logra cierta motivación tanto en las personas que los donan como en los nuevos estudiantes, quienes se animan y se van haciendo la idea de dejar proyectos en el laboratorio en el futuro. Además, hay que agregar que esto beneficia a la universidad, pues desde el punto de vista económico supone un ahorro, y desde el punto de vista estudiantil favorece tanto a maestros como a alumnos.

## Anexo 2. Fotos del Laboratorio de Automatización



Figura 73. Brazo robótico.

Fuente: Autor.

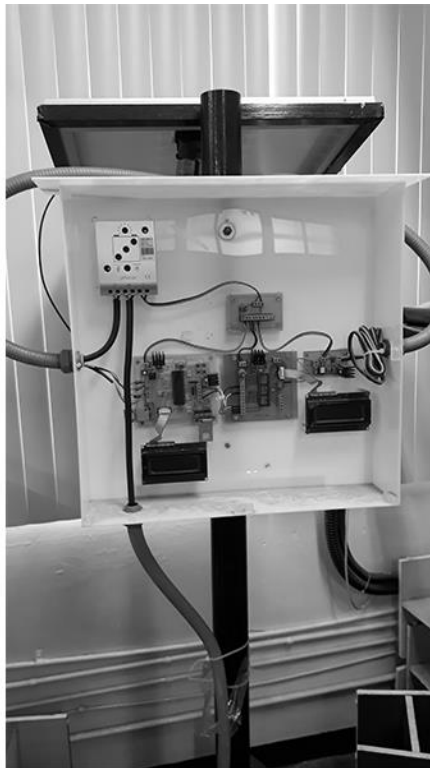


Figura 74. Sistema de riego.

Fuente: Autor.



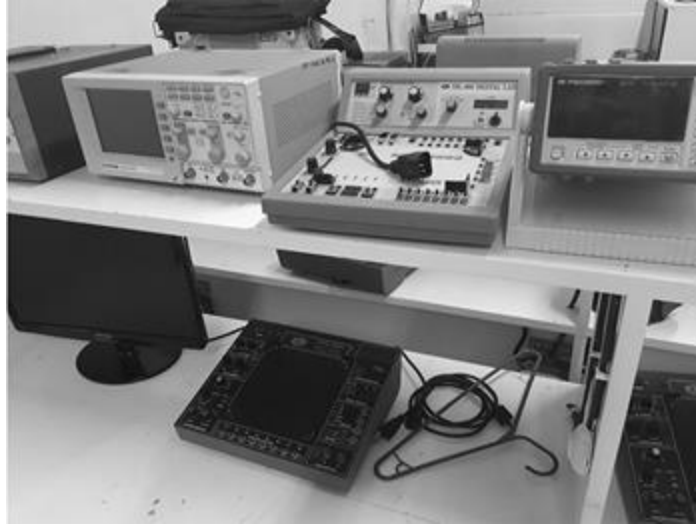


Figura 75. Equipos de medición.

Fuente: Autor.



Figura 76. Boiler.

Fuente: Autor.

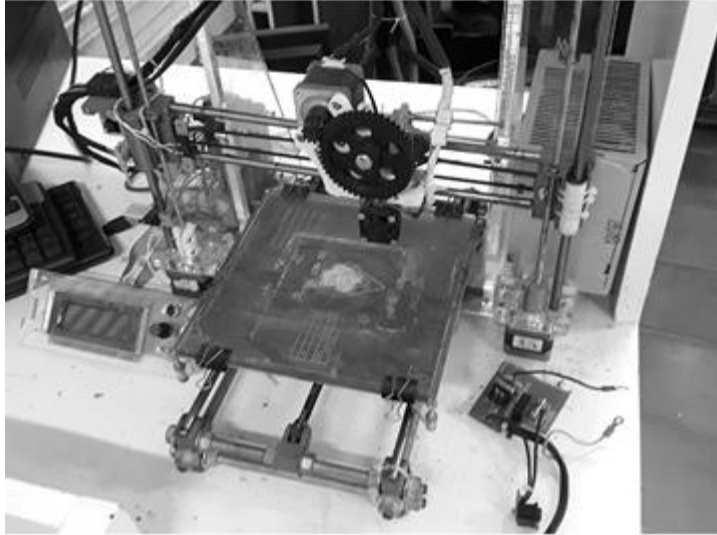


Figura 77. Impresora 3D.

Fuente: Autor.

### Anexo 3. YUMO Rotary Encoder Datasheet (A6B2-CWZ3E-1024)

# YUMO

## 旋转编码器 ROTARY ENCODERS

### 增量型旋转编码器

增量型 外径 $\phi 40$  型号: A6B2

INCREMENTAL ROTARY ENCODERS, OUTSIDE DIAM  $\phi 40$  MODEL: A6B2

替代型号 SUBSTITUTE: E6B2

### 通用编码器

#### 型号

**A6B2 - CWZ 6C**

设计序号  
Sequence Number

外径 $\phi 40$   
outside diam  $\phi 40$

1: DC5V  
3: DC5~12V  
5: DC12~24V  
6: DC5~24V

Z: 带复位相输出 + Zero Signal  
S: 单相输出 singleness Output "A"  
W: 多相输出 A、B 90° Phase Difference

A: 绝对式编码器 Absolute Rotary Encoder  
C: 增量式编码器 Incremental Rotary Encoder

B: PNP开路输出 PNP open collector Output  
C: NPN开路输出 Open collector NPN output  
E: 电压 (NPN) 输出 Voltage output  
Gh: 互补输出 NPN Push Pull output  
EH: 电压 (PNP) 输出 PNP Voltage output

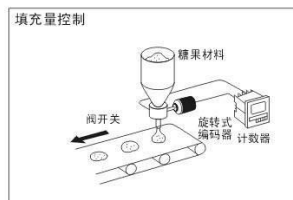
#### ■ 特点 CHARACTERISTIC

- 对应宽电源电压DC5~24V (开路集电极型)
- 外径 $\phi 40$ 备有到2000P/R的高分辨率。
- 具备使Z相对合简化的原点位置显示功能。
- 轴负重、径向29.4N, 推力向19.6N, 允许大幅度提升。
- 附有逆接、负荷短路保护电路, 改善了可靠性。
- 也备有线性驱动输出。  
最大可延长长度100m为止

General-Purpose Rotary Encoder  
Withstands Large Shaft Loads

- Wide variety of supply voltages and output forms
- Easy-to-adjust zero index (phase Z) with origin indicating function High resolution models (2000 pulses per revolution) substantially improves measuring accuracy
- Rugged construction: 6 mm (0.24 inch) diameter shaft with radial load ratings of 3 kgf (21.7 ft\*lbs) and axial load rating of 2 kgf (14.5 ft\*lbs)
- Protected against short-circuit and reversed connections for highly reliable operation Available with Line Driver output

#### ■ 应用实例 APP-example

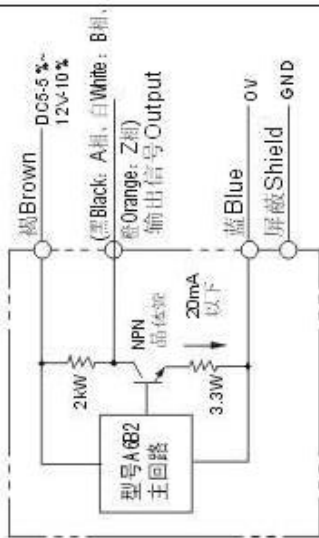


YUEQING YING'S IMPORT & EXPORT CO.,LTD [WWW.YINGSELECTRIC.COM](http://WWW.YINGSELECTRIC.COM)

## 额定/性能SPECIFICATIONS

项目 ITEM	型号 MODEL	型号A6B2-CWZ5C	型号A6B2-CWZ5B	型号A6B2-CWZ3E	型号A6B2-CWZ1X
电源电压 Powersupply voltage		DC5V-5%~24V+15% 脉冲(p-p)5%以下	<b>DC12V</b> -10%~24V+15% 脉冲(p-p)5%以下	DC5V-5%~12V+10% 脉冲(p-p)5%以下	DC5V±5% 脉冲(p-p)5%以下
消耗电流 Currentconsumption		80mA以下	100mA以下	100mA以下	160mA以下
分辨率(脉冲/旋转) Resolution(See Note 1)		10、20、30、40、50、60、100、200、300、360、400、500、600、1,000、1,200、1,500、1,800、2,000、	100、200、360、500、600、1,000、2,000	10、20、30、40、50、60、100、200、300、360、400、500、600、1,000、1,200、1,500、1,800、2,000、	
输出相 Output phases		A、B、Z相			A、 $\bar{A}$ 、B、 $\bar{B}$ 、Z、 $\bar{Z}$ 相
输出相位差 Phase difference of output		A相、B相的相位差 $90 \pm 45^\circ$ ( $14 \pm 18^\circ$ )			
输出状态 Output form		NPN开路输出 Open collectorNPN output	PNP开路输出 Open collector PNP output	电压输出 Voltage output	线性驱动输出*2 Line driver output
输出容量 Output capacity		外加电压: DC30V以下 In Voltage: DC30V max 同步电流: 35mA以下 In-sink: 35 mA max 残留电压: 0.4V以下 Residual voltage: 0.4 V max (输出电流) 35mA时 (In-sink: 35mA )	同步电流: 35mA以下 In-sink: 35mA max 残留电压: 0.4V以下 Residual voltage: 0.4 V max (输出电流) 35mA时 (In-sink: 35mA )	输出电阻: 2k $\Omega$ Output resistance: 2 k $\Omega$ 输出电流: 35mA以下 Sink current: 35 mA max 残留电压: 0.4V以下 Residual voltage: 0.4 V max (输出电流) 35mA时	AM26LS31相当品 输出电流 H位: Io=20mA High level L位: Is=20mA 输出电压 VO=2.5V以上min Low level VS=0.5V以下max
最高应答频率 Maximum response frequency		100kHz	50kHz	100kHz	
输出上升、下降时间 Output rise and fall times		1ms以下 (控制输出电压: 5V 负载电阻1k $\Omega$ 、导线长: 2m)	1ms以下 (导线长: 2m 同步电流: 10mA)	1ms以下 (导线长: 2m Io=-20mA、Is=20mA)	

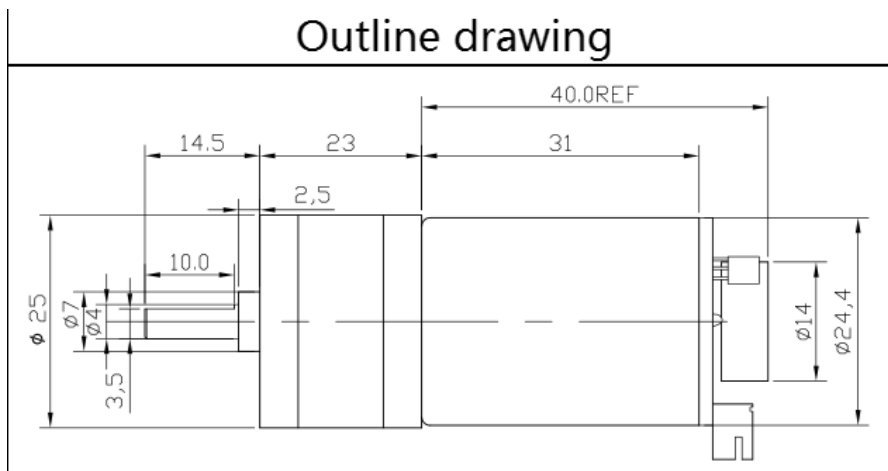
输出段回路图OUTPUT CIRCUIT DIAGRAMS

型号/输出回路 MODEL/OUTPUT	输出方式 Output wave	连接 Wire color code												
型号A6B2-CWZ3E	<div><p>型号A6B2主回路</p><p>2kW</p><p>3.3W</p><p>20mA</p><p>以下</p><p>输出信号 Output</p><p>屏蔽 Shield</p><p>GND</p><p>DC5.5V 12V-10V</p><p>褐Brown</p><p>黑Black: A相, 白White: B相, 橙Orange: Z相</p></div> <p>电压输出/型号A6B2-CWZ3E Voltage Output /A6B2-CWZ3E</p> <div><p>旋转方向: CW Direction of rotation: CW (从轴方向观察为右转)</p><p>晶体管输出</p><p>A相</p><p>B相</p><p>Z相</p><p>CW方向</p><p>晶体管输出</p><p>A相</p><p>B相</p><p>Z相</p><p>CCW方向</p><p>注: A相比B相前<math>1/4 \pm 1/8T</math>相位。 Note: Phase A is <math>1/4T \pm 1/8T</math> faster than phase B.</p><p>注: A相比B相滞后<math>1/4 \pm 1/8T</math>相位。 Note: Phase A is <math>1/4T \pm 1/8T</math> slower than phase B.</p></div>	<p>Wire Color Code IEC colors are shown.</p> <table><tr><th>线色Wire color</th><th>端子名Signal</th></tr><tr><td>褐Brown</td><td>电源(+Vcc)</td></tr><tr><td>黑Black</td><td>输出A相</td></tr><tr><td>白White</td><td>输出B相</td></tr><tr><td>粉Orange</td><td>输出Z相</td></tr><tr><td>蓝Blue</td><td>0V (COMMON)</td></tr></table>	线色Wire color	端子名Signal	褐Brown	电源(+Vcc)	黑Black	输出A相	白White	输出B相	粉Orange	输出Z相	蓝Blue	0V (COMMON)
线色Wire color	端子名Signal													
褐Brown	电源(+Vcc)													
黑Black	输出A相													
白White	输出B相													
粉Orange	输出Z相													
蓝Blue	0V (COMMON)													

#### Anexo 4. 12V 350RPM DC Gear Motor with Hall Encoder Datasheet



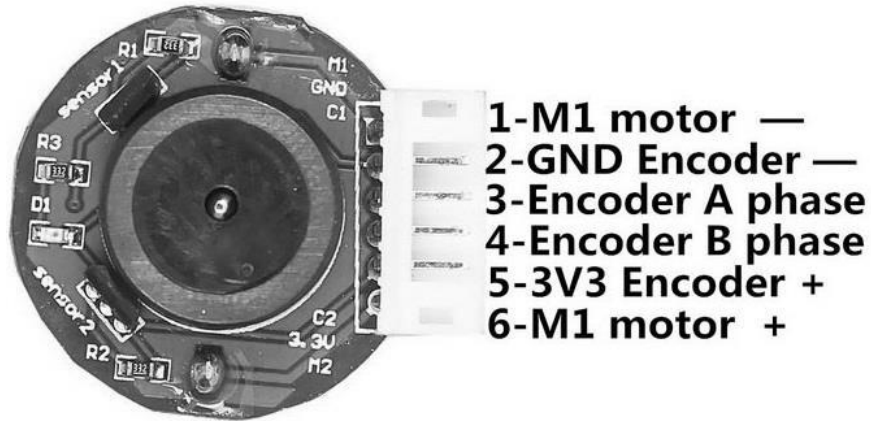
Outline drawing



##### Specification:

Voltage	DC 12.0V
Speed	350RPM
Shaft diameter	4mm
Length	12mm
Encoder motor end	11 signals
Terminal connection length	20cm

Rated Voltage	DC 12V
No-load Speed	350RPM 0.2A
Max Efficiency	2.0kg.cm/285rpm/5.0W/0.65A
Max Power	5.8kg.cm/180rpm/9.0W/1.65A
Stall Torque	12kg.cm 5.5A
Retarder Reduction Ratio	1:34
Hall Resolution	Hall 11 x Ratio 34 = 374 PPR



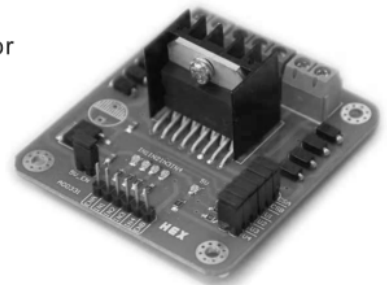
## Anexo 5. L298N H-Bridge Motor Driver Module Datasheet

### Overview

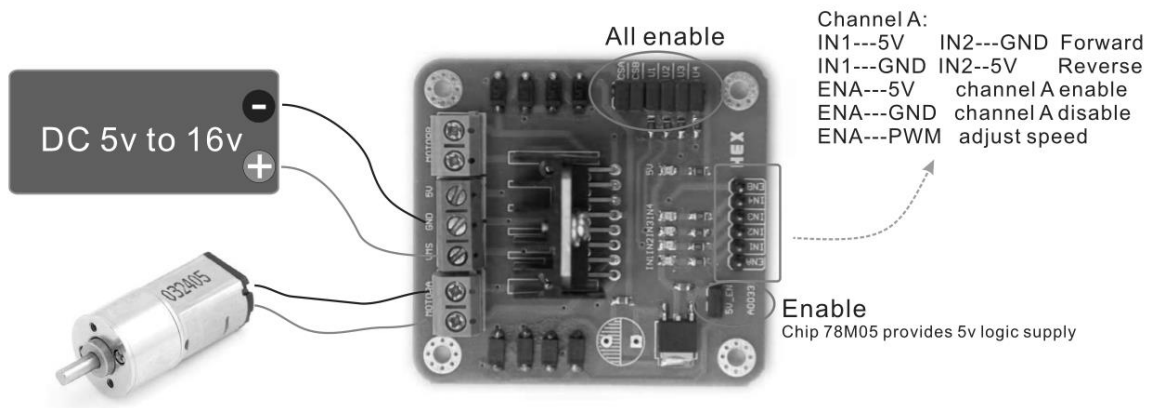
The Motor Shield is based on the L298, which is a dual full-bridge driver designed to drive inductive loads such as relays, solenoids, DC and stepping motors. It lets you drive two DC motors , controlling the speed and direction of each one independently.







### Summary

Operating Voltage 4V to 35V  
 Motor controller L298N, Drives 2 DC motors or 1 stepper motor  
 Max current 2A per channel or 4A max  
 Free running stop and brake function  
 Chip: ST L298N  
 Logic power supply:5v  
 Max power:25w  
 Weight: 35g  
 Size:55mm x 60mm x 30mm  
 Storage temperature:-25 to +135



DC brush motor



	<p>CSA: Between this pin and ground is connected the sense resistor to control the current of the load. Enable----- Ignore current detection function</p> <p>CSB: Between this pin and ground is connected the sense resistor to control the current of the load. Enable----- Ignore current detection function</p>	
	<p>5V-EN: Enable----78M05 worked ,output DC 5V Disable----78M05 do not work . Need input DC 5V The module need DC 5V always, for logic supply.</p>	
	<p>The pull-up resistor enabled. U1---Enable In1 pull-up resistor [10k]. U2---Enable In2 pull-up resistor [10k]. U3---Enable In3 pull-up resistor [10k]. U4---Enable In4 pull-up resistor [10k].</p>	
	<p>IN1 IN2 :TTL Compatible Inputs of the Bridge A IN3 IN4 :TTL Compatible Inputs of the Bridge B. ENA ENB:TTL Compatible Enable Input: the L state disables the bridge A(enable A) and/or the bridge B (enable B).</p>	<p>5V_EN: Enable: [5V] can output DC 5V. Disable:[5v] need input DC 5V.</p>

## Anexo 6. Scripts de MATLAB

### Pendulum\_identification.m

```
%-----
%   Pendulum_identification.m
%-----
%%      Description
%
% This is a script to identify the parameters of the pendulum
% system from experimental data.
%
%%      Output
%
% Pendulum ODE and its solution.
% Figures:
%   -Pendulum Free Response
%   -Pendulum Parameters Identification: Plot of the
%     Experimental data
%     vs. Estimated model
%   -Pendulum roots in the complex plane Estimated Parameters:
%   -Pendulum ODE
%   -Pendulum ODE Solution
%
% In the program section, to use and save data from workspace
```



```

% comment load and uncomment save. To use loaded data from
% pendulum.mat uncomment load and comment save.
%
%%      Version
%
% Author: Felix Ruiz Torrez (torrezfelix@gmail.com)
% Creation date: 28-dec-2018
% Matlab version: R2018a
%
%%      Revision
%
% V1.0 | 28-dec-2018 | Felix Ruiz Torrez | Creation

%%      Program
clc
close all

load('pendulum.mat'); %load the experimental data of the pendulum
%save('pendulum.mat','pdata'); %save experimental data

%% 1.) Definition
%% 1.) - Parameter definition
angle_pulses = squeeze(pdata.signals.values); %angle in pulses
time = pdata.time; %time in seconds
encoder_resolution = 1024;
encoder_counting = 4;
encoder2radians = 2*pi/(encoder_resolution*encoder_counting);
angle_radians = angle_pulses*encoder2radians; %angle in radians
[pks,locs] = findpeaks(angle_radians); %find all the maximas
y1 = pks(2); %First maxima
t1 = time(locs(2)); % time of first maxima
y2 = pks(3); %Second maxima
t2 = time(locs(3)); %time of second maxima
timemodel = time(locs(2):end)-t1; %model time

%% 2.) Computing
%% 2.) - ODE parameter estimation
alpha = -(1/(t2-t1))*log(y2/y1); %damping factor
tau = 1/alpha; %time constant
omega_d = 2*pi/(t2-t1); %damped angular velocity
omega_0 = sqrt((omega_d^2)+(alpha^2)); %undamped angular velocity
theta_0 = y1; %theta initial condition
thetadot_0 = 0; %theta_dot initial condition
s1 = -alpha+1i*omega_d; %pendulum first root
s2 = -alpha-1i*omega_d; %pendulum second root
k1 = thetadot_0-s2*theta_0/(s1-s2); %constant of the first root
k2 = thetadot_0-s1*theta_0/(s2-s1); %constant of the second root
beta = atan(abs(imag(k1))/abs(real(k1))); %cosine phase in rad
envelope = 2*abs(k1)*exp(-alpha*timemodel); %free response envel.

%% 2.) - General solution to ODE
theta_sol = k1*exp(s1*timemodel)+k2*exp(s2*timemodel);

```

```

%% 2.) - Symbolic ODE
syms t theta theta_dot theta_dot_dot;
edo = theta_dot_dot+double(2*alpha)*theta_dot+omega_0^2*theta==0;
edo_p = vpa(edo,4);

%% 2.) - Pendulum ODE Solution (phase in radians)
theta_sol_pretty = 2*abs(k1)*exp(-
alpha*t)*cos(omega_d*t+beta)==theta;
theta_p = vpa(theta_sol_pretty,4);

%% 2.) - Pendulum transfer function used to plot the roots
PendulumTf = tf(1,[1,(2*alpha),omega_0^2]);

%% 3.) Plotting
%% 3.) - Pendulum Experiment graph
figure('Name','Pendulum Experiment','NumberTitle','off');
plot(time,angle_radians,'k',t1,y1,'*k',t2,y2,'*k'); %Experiment
plot
legend('experiment');
xlabel('time (sec)');
ylabel('angle (radians)');
title('Pendulum Free Response');
set(gcf,'color','w');

%% 3.) - Pendulum Model validation graph
figure('Name','Pendulum Model Validation','NumberTitle','off');
plot(time,angle_radians,'k'); %plot experiment data
hold on; %hold the plot
plot(timemodel+t1,theta_sol, ...
'ok','MarkerIndices',1:5:length(theta_sol)); %plot model
solution
plot(timemodel+t1,-envelope,'k:'); %positive envelope plot
plot(timemodel+t1,-envelope,'k:'); %negative envelope plot
hold off;
legend({'experiment','model','envelope'});
xlabel('time (sec)');
ylabel('angle (radians)');
title('Pendulum Free Response');
xlim([t1, time(locs(7))]); %limit x axis
ylim([-1.2*y1,1.4*y1]); %limit y axis
set(gcf,'color','w');

%% 3.) - Pendulum Roots graph
figure('Name','Pendulum Roots','NumberTitle','off');
pzplot(PendulumTf,'k');
title('Pendulum Roots');
xlim([-1.2*alpha, 0]);
ylim([-1.2*omega_d 1.2*omega_d]);
h1 = findobj(gca, 'type', 'line');
set(h1, 'markersize', 15,'color','k'); %increase marker size
set(gcf,'color','w'); %set background color to white

```

```

%% 4.) Display
%% 4.) - Pendulum ODE
fprintf('Pendulum ODE = \n\n');
pretty(edo_p);

%% 4.) - Pendulum ODE solution
fprintf('Pendulum ODE Solution (phase in radians) = \n\n');
pretty(theta_p);

```

### **Dc\_motor\_identification.m**

```

%-----
%   Dc_motor_identification.m
%-----
%%       Description
%
%   This is a script to identify the parameters of the Cart from
%   experimental data.
%
%%       Output
%
% Estimated model:
%   - motor_tf: motor transfer function
% Estimated Parameters:
%   - Km: motor gain.
%   - Tm: motor time constant.
%   - motor_tf: motor transfer function.
% Figures:
%   - Experiment.
%   - Model validation.
%   - Cart roots.
%
%%       Version
%
% Author: Felix Ruiz Torrez (torrezfelix@gmail.com)
% Creation date: 30-dec-2018
% Matlab version: R2018a
%
%%       Revision
%
% V1.0 | 30-dec-2018 | Felix Ruiz Torrez | Creation

%%       Program
clc;
close all;

load ('motor.mat');
%save('motor.mat','mdata');

```

```

%% 1.) Definition
%% 1.) - Parameter definition
u = squeeze(mdata.signals.values(:,1,:));
y = squeeze(mdata.signals.values(:,2,:));
t = mdata.time;
ui = cumtrapz(t,u);
ui2 = cumtrapz(t,ui);
yi = cumtrapz(t,y);
yi2 = cumtrapz(t,yi);

%% 2.) Computing
%% 2.) - Parameter estimation
p = [y,yi]\ui2; %back slash operator to obtain the vector p
Km = 1/p(2); %motor gain
Tm = p(1)/p(2);%motor time constant

%% 2.) - Motor transfer function
motor_tf = tf(Km,[Tm,1,0], ...
    'InputName','pwm','OutputName','Motor Shaft Position
    (pulses)');

%% 2.) - Motor simulation
motor_sim = lsim(motor_tf,u,t);

%% 3.) Plotting
%% 3.) - Experiment graph
fig = figure('Name','Motor Experiment','NumberTitle','off');
left_color = [0 0 0];
right_color = [0 0 0];
set(fig,'defaultAxesColorOrder',[left_color; right_color]);
yyaxis right
plot(t,y,'k','MarkerIndices',1:15:length(y));
ylabel('motor shaft position (pulses)');
yyaxis left
plot(t,u,'s:k','MarkerIndices',1:15:length(u),'MarkerSize',3.5);
xlabel('time (sec)');
ylabel('pwm');
title('Motor Input - Output');
legend('pwm','shaft position','Location','southeast');
set(gcf,'color','w');

%% 3.) - Model validation graph
figure('Name','Motor Model Validation','NumberTitle','off');
plot(t,y,':ok','MarkerIndices',1:15:length(y));
hold on;
plot(t,motor_sim,'k');
title('Motor Response');
legend('experiment','estimated model','Location','southeast');
xlabel('time (sec)');
ylabel('motor shaft position (pulses)');
set(gcf,'color','w');

```

```

%% 3.) - Motor roots
figure('Name','Motor Roots','NumberTitle','off');
pzplot(motor_tf,'k');
title('Motor Roots');
h1 = findobj(gca, 'type', 'line');
set(h1, 'markersize', 15, 'color', 'k'); %increase marker size
set(gcf, 'color', 'w'); %set background color to white

%% 4.) Display
%% 4.) - Estimated parameters
display(Km); %motor gain
display(Tm); %motor time constant

%% 4.) - Motor Transfer Function
display(motor_tf); %motor transfer function

```

### **Cart\_Pendulum\_identification.m**

```

%-----
%   Cart_Pendulum_identification.m
%-----
%%       Description
%
%   This is a script to identify the parameters of the pendulum
%   and inverted pendulum on the cart systems (pendulum down and
%   pendulum up)
%
%%       Output
%
% Estimated Models:
%   - sspendown= Continuous state space model pendulum down on
%   the cart
%   - sspendownd= Discrete state space model pendulum down on the
%   cart
%   - sspendup= Continuous state space model inverted pendulum on
%   the cart
%   - sspendupd= Discrete state space model inverted pendulum on
%   the cart
%   - cartpend_tf= Transfers functions of the cart, pendulum
%   up/down on the cart
%
% Figures:
%   - Pendulum on the Cart Experiment (pendulum down)
%   - Pendulum on the Cart Model Validation (pendulum down)
%
%%       Version
%
% Author: Felix Ruiz Torrez (torrezfelix@gmail.com)
% Creation date: 3-jan-2019
% Matlab version: R2018a

```

```

%
%%      Revision
%
% V1.0 | 3-jan-2019 | Felix Ruiz Torrez | Creation

%%      Program
clc
close all

load('ipdata.mat') %comment this line to use data from workspace
%save('ipdata.mat','ipdata');

Ts = 0.005; %system sample time

%% 1.) Definition
%% 1.) - Pendulum parameter definition
angle_pulses = squeeze(ipdata.signals.values(:,3,:));
t = ipdata.time;
encoder_resolution = 1024;
encoder_counting = 4;
encoder2radians = 2*pi/(encoder_resolution*encoder_counting);
%encoder pulses to radians
angle_radians = angle_pulses*encoder2radians; %angle in radians
[pks,locs] = findpeaks(angle_pulses); %find the maximas
y1 = pks(2); %second maxima
t1 = t(locs(2)); %time of the second maxima
y2 = pks(3); %Third maxima
t2 = t(locs(3)); %time of the third maxima
a = 1; %initial guess a=1

%% 1.) - Cart parameter definition
u = squeeze(ipdata.signals.values(:,1,:));
y = squeeze(ipdata.signals.values(:,2,:));
ui=cumtrapz(t,u);
ui2=cumtrapz(t,ui);
ui3=cumtrapz(t,ui2);
yi=cumtrapz(t,y);
yi2=cumtrapz(t,yi);
yi3=cumtrapz(t,yi2);

%% 2.) Computing
%% 2.) - Cart parameter estimation
p=[y,yi]\ui2; %p vector, characteristic polynomial
Km = 1/p(2); %motor gain
Tm = p(1)/p(2);%motor time constant

%% 2.) - Pendulum parameter estimation
alpha = -(1/(t2-t1))*log(y2/y1);
tau = 1/alpha;
omega_d = 2*pi/(t2-t1);
omega_0 = sqrt(omega_d^2+alpha^2);

```

```

%% 2.) - Continuous ss of the pendulum on the cart with a=1
A = [0,1,0,0;
      0,-(p(2)/p(1)),0,0;
      0,0,0,1;
      0,a*(p(2)/p(1)),-(omega_0^2),-2*alpha];
B = [0;
      1/p(1);
      0;
      -a/p(1)];
C = eye(4);
D = [0;0;0;0];
sspendown = ss(A,B,C,D);
[~,~,ssxa] = lsim(sspendown,u,t);

%% 2.) - Parameter "a" identification (adjust pendulum amplitude)
[ss_pks,ss_locs] = findpeaks(ssxa(:,3));
a = pks(2)/ss_pks(2);

%% 2.) - Continuous ss model with estimated parameter "a"
A = [0,1,0,0;
      0,-(p(2)/p(1)),0,0;
      0,0,0,1;
      0,a*(p(2)/p(1)),-(omega_0^2),-2*alpha];
B = [0;
      1/p(1);
      0;
      -a/p(1)];
states = {'x' 'x_dot' 'theta' 'theta_dot'};
inputs = {'u (pwm)'};
outputs = {'x';'x_dot';'theta';'theta_dot'};
sspendown =
ss(A,B,C,D,'statename',states,'inputname',inputs,'outputname',out
puts);
[~,~,ssx] = lsim(sspendown,u,t);

%% 2.) - Continuous ss model of the inverted pendulum on the cart
AI = [0,1,0,0;
       0,-(p(2)/p(1)),0,0;
       0,0,0,1;
       0,a*(p(2)/p(1)),(omega_0^2),-2*alpha];
BI = [0;
       1/p(1);
       0;
       -a/p(1)];
sspendup =
ss(AI,BI,C,D,'statename',states,'inputname',inputs,'outputname',o
utputs);

%% 2.) - Discrete ss models of the pendulum down & up on the cart
sspendownd = c2d(sspendown,Ts);
sspendupd = c2d(sspendup,Ts);

```

```

%% 2.) - Pendulum down & up transfer functions
pendowntf = tf([-a/p(1),0,0], ...
    [p(1), (p(2)+2*alpha*p(1)), ...
    (2*alpha*p(2)+(omega_0^2)*p(1)), ...
    (omega_0^2)*p(2),0]/p(1)); %pendulum on the cart
penduptf = tf([-a/p(1),0,0], ...
    [p(1), (p(2)+2*alpha*p(1)), ...
    (2*alpha*p(2)-(omega_0^2)*p(1)), ...
    -1*(omega_0^2)*p(2),0]/p(1)); %inverted pendulum on the cart

%% 2.) - Cart Transfer function
cart_tf = tf(Km,[Tm,1,0]);

%% 2.) - Open Loop System Impulse Response plot
sys_tf = [cart_tf ; penduptf];
inputs = {'u'};
outputs = {'x'; 'theta'};
set(sys_tf,'InputName',inputs);
set(sys_tf,'OutputName',outputs);

%% 3.) Plotting
%% 3.) - Experiment graph
figure('Name','Pendulum on the Cart Experiment',
    'NumberTitle','off');
subplot(3,1,1);
plot(t,u,'k');
xlabel('time (sec)');
ylabel('pwm');
ylim([min(u)*1.2,max(u)*1.2]);
title('Motor Input');
subplot(3,1,2);
plot(t,y,'k');
xlabel('time (sec)');
ylabel({'cart position'; '(pulses)'});
ylim([min(y)*1.2,max(y)*1.2]);
title('Cart Response');
subplot(3,1,3);
plot(t,angle_pulses,'k');
xlabel('time (sec)');
ylabel({'pendulum angle'; '(pulses)'});
ylim([-max(angle_pulses)*1.2,max(angle_pulses)*1.2]);
title('Pendulum Response');
set(gcf,'color','w');

%% 3.) - Cart model validation graph, "a" estimation
figure('Name','Parameter "a" Estimation','NumberTitle','off');
subplot(211);
plot(t,y,'k');
hold on;
plot(t,ssx(:,1),':ok','MarkerIndices',1:floor(0.3/Ts):length(y));
hold off;
legend('experiment','model','Location','southeast');

```



```

xlabel('time (sec)');
ylabel('cart position (pulses)');
title('Cart Response');
xlim([t(locs(1))-1.5,t(locs(5))]);
ylim([0,max(y)*1.2]);

%% 3.) - Pendulum model validation graph, "a" estimation
subplot(212);
plot(t,angle_pulses,'k');
hold on; %hold the plot
plot(t,(ssxa(:,3)),':ok','MarkerIndices',1:floor(0.1/Ts):length(ssx(:,3)));
plot(t1,y1,'*k');
plot(t(ss_locs(2)),ss_pks(2),'*k')
hold off;
legend({'experiment','model'});
xlabel('time(sec)');
ylabel('pendulum angle (pulses)');
title('Pendulum Response');
xlim([t(locs(1))-1.5,t(locs(5))]);
ylim([-max(ssxa(:,3))*1.2,max(ssxa(:,3))*1.2]);
set(gcf,'color','w');

%% 3.) - Cart model validation graph
figure('Name','Pendulum on the Cart Model Validation','NumberTitle','off');
subplot(211);
plot(t,y,'k');
hold on;
plot(t,ssx(:,1),':ok','MarkerIndices',1:floor(0.3/Ts):length(y));
hold off;
legend('experiment','model','Location','southeast');
xlabel('time (sec)');
ylabel('cart position (pulses)');
title('Cart Response');
xlim([t(locs(1))-1.5,t(locs(5))]);
ylim([0,max(y)*1.2]);

%% 3.) - Pendulum model validation graph
subplot(212);
plot(t,angle_pulses,'k');
hold on; %hold the plot
plot(t,(ssx(:,3)),':ok','MarkerIndices',1:floor(0.1/Ts):length(ssx(:,3)));
hold off;
legend({'experiment','model'});
xlabel('time(sec)');
ylabel('pendulum angle (pulses)');
title('Pendulum Response');
xlim([t(locs(1))-1.5,t(locs(5))]);
ylim([-max(angle_pulses)*1.2,max(angle_pulses)*1.2]);
set(gcf,'color','w');

```

```

%% 3.) - Inverted Pendulum Open Loop impulse response
figure('Name','Inverted Pendulum Open Loop Impulse Response',...
      'NumberTitle','off');
impz(sys_tf,'k',1);
title('Open Loop Impulse Response');
set(gcf,'color','w'); %set background color to white

%% 4.) Display
%% 4.) - Continuous ss pendulum (down & up) on the cart
display(sspendown); %continuous time
display(sspendup); %continuous time

%% 4.) - Discrete ss pendulum (down & up) on the cart
display(sspendownd); %discrete time
display(sspendupd); %discrete time

%% 4.) - Inverted Pendulum on the cart transfer function
display(penduptf);

```

### **Inverted\_Pendulum\_PID\_design.m**

```

%-----
%   Inverted_Pendulum_PID_design.m
%-----
%%      Description
%
%   This is a script to compute the model based PID controller
%   design of the inverted pendulum on the cart using Evan's root
%   locus method.
%
%%      Output
%
%   Controllers:
%   - cpitf = Continuous time PI
%   - gczpi = Digital PI
%   - cpidtf = Continuous time PID
%   - gczpid = Digital PID
%
%   Figures:
%   -Root Locus with positive K
%   -Root Locus with negative K
%   -Root Locus with PI control
%   -Root Locus with PID control
%   -Impulse disturbance response of pendulum angle under PI
%   control
%   -Impulse disturbance response of cart position under PI
%   control
%   -Impulse disturbance response of pendulum angle under PID
%   control

```

```

% -Impulse disturbance response of cart position under PID
% control
%
%% Version
%
% Author: Felix Ruiz Torrez (torrezfelix@gmail.com)
% Creation date: april-2019
% Matlab version: R2018a
%
%% Revision
%
% V1.0 | 1-mar-2019 | Felix Ruiz Torrez | Creation

%% Program
run('Cart_Pendulum_identification.m');
close all;
clc;

Tst=0.03; %Implementation transmission sample time
MotorDeadZone=35; %Implementation motor dead zone

%% 1.) Definition
%% 1.) - Inverted pendulum transfer function
rpenduptf=minreal(penduptf); %reduced inverted pendulum tf
rpenduptfd=c2d(penduptf,Ts); %discrete inverted pendulum tf

%% 1.) - Cart transfer function
cart_tfd=c2d(cart_tf,Ts); %discrete cart transfer function

%% 1.) - PI gains
k = -1;
p = 0;
z_pi = -3; %zero at -3
kpi = 7; %root locus gain. kpi equivalent to P gain for PI

%% 1.) - PID gains
z_pid = [-12 -15]; %zeros due to P and D gains
kpid = 4; %root locus gain. kpid is equivalent to D gain for PID

%% 2.) Computing
%% 2.) - Analog PI
cpi = zpk(z_pi,p,k); %zero-pole PI controller
cpitf=tf(kpi*cpi); %analog PI controller

%% 2.) - Analog PID
cpid = zpk(z_pid,p,k); %PID controller zero pole
cpidtf=tf(kpid*cpid); %analog PID controller

%% 2.) - Digital PI
kp_pi=cpitf.Numerator{1,1}(1); %proportional gain analog PI
ki_pi=cpitf.Numerator{1,1}(2); %integral gain analog PI
kpd_pi=kp_pi-ki_pi*Ts/2; %proportional gain digital PI

```

```

kid_pi=ki_pi*Ts;      %integral gain digital PI
gczpi = tf([kpd_pi+kid_pi -kpd_pi],[1 -1],Ts); %digital PI

%% 2.) - Digital PID
kp_pid=cpidtf.Numerator{1,1}(2);      %proportional gain analog PID
ki_pid=cpidtf.Numerator{1,1}(3);      %integral gain analog PID
kd_pid=cpidtf.Numerator{1,1}(1);      %derivative gain analog PID
kpd_pid=kp_pid-ki_pid*Ts/2;           %proportional gain digital PID
kid_pid=ki_pid*Ts; %integral gain digital PID
kdd_pid=kd_pid/Ts; %derivative gain digital PID
gczpid = tf([kpd_pid+kid_pid+kdd_pid -kpd_pid-2*kdd_pid
kdd_pid],[1 -1 0],Ts); %digital PID

%% 2.) - PI root locus & close loop
rlpi=minreal(cpi*rpenduptf); %root locus
pipend = feedback(rpenduptf,cpitf); %continuous pendulum response
pipendd = feedback(rpenduptfd,gczpi); %discrete pendulum response
picart = feedback(1,rpenduptf*cpitf)*cart_tf; %cont cart response
picartd = feedback(1,rpenduptfd*gczpi)*cart_tfd; %discrete cart
response

%% 2.) - PID root locus & close loop
rlpid=minreal(cpid*rpenduptf); %root locus
pidpend = feedback(rpenduptf,kpid*cpid); %cont. pendulum response
pidpendd = feedback(rpenduptfd,gczpid); %disc. pendulum response
pidcart = feedback(1,rpenduptf*cpidtf)*cart_tf; %continuous cart
response
pidcartd = feedback(1,rpenduptfd*gczpid)*cart_tfd; %discrete cart
response

%% 3.) Plotting
%% 3.) - Root Locus with positive K graph
figure('Name','Root Locus with positive K','NumberTitle','off');
rlocus(rpenduptf)
xlim([-12 9]);
title('Root Locus with positive K');
h1 = findobj(gca, 'type', 'line');
set(h1, 'markersize', 15, 'color', 'k');
set(gcf, 'color', 'w');

%% 3.) - Root Locus with negative K graph
figure('Name','Root Locus with negative K','NumberTitle','off');
rlocus(-rpenduptf)
%xlim([-15 15]);
title('Root Locus with negative K');
h2 = findobj(gca, 'type', 'line');
set(h2, 'markersize', 15, 'color', 'k');
set(gcf, 'color', 'w');

%% 3.) - Root Locus PI control graph
figure('Name','Root Locus PI Control','NumberTitle','off');
rlocus(rlpi);

```

```

%xlim([-15 15]);
title('Root Locus with PI Control');
h3 = findobj(gca, 'type', 'line');
set(h3, 'markersize', 15, 'color', 'k');
set(gcf, 'color', 'w');

%% 3.) - Root Locus PID control graph
figure('Name', 'Root Locus PID Control', 'NumberTitle', 'off');
rlocus(rlpid);
xlim([-40 10]);
title('Root Locus with PID Control');
h3 = findobj(gca, 'type', 'line');
set(h3, 'markersize', 15, 'color', 'k');
set(gcf, 'color', 'w');

%% 3.) - Pendulum Impulse Response under PI control graph
figure('Name', 'Pendulum Impulse Response under PI
control', 'NumberTitle', 'off');
impz(pipend, 'k');
hold on;
[ipd, tipd] = impz(pipendd);
plot(tipd, ipd, ':ok', 'MarkerIndices', 1:20:length(tipd), 'MarkerSize',
3);
hold off;
xlim([tipd(1) tipd(end)]);
legend('Analog PI', 'Digital PI');
title('Impulse Disturbance Response of Pendulum Angle under PI
Control');
set(gcf, 'color', 'w');

%% 3.) - Cart Impulse Response under PI control graph
figure('Name', 'Cart Impulse Response under PI
control', 'NumberTitle', 'off');
impz(picart, 'k');
hold on;
[ipicd, tipicd] = impz(picartd);
plot(tipicd, ipicd, ':ok', 'MarkerIndices', 1:50:length(tipicd), 'Mark
erSize', 3);
hold off;
xlim([tipicd(1) tipicd(end)]);
legend('Analog PI', 'Digital PI');
title({'Impulse Disturbance Response of Cart Position', 'under PI
Control'});
set(gcf, 'color', 'w');

%% 3.) - Pendulum Impulse Response under PID control graph
figure('Name', 'Pendulum Impulse Response under PID
control', 'NumberTitle', 'off');
impz(pidpend, 'k');
hold on;
[ippd, tippd] = impz(pidpendd);

```

```

plot(tippd,ippd,':ok','MarkerIndices',1:3:length(tippd),'MarkerSi
ze',3);
hold off;
xlim([tippd(1) tippd(end)]);
legend('Analog PID','Digital PID');
title('Impulse Disturbance Response of Pendulum Angle under PID
Control');
set(gcf,'color','w');

%% 3.) - Cart Impulse Response under PID control graph
figure('Name','Cart Impulse Response under PID
control','NumberTitle','off');
impulse(pidcart,'k');
hold on;
[tipidcd,ipidcd]=impulse(pidcartd);
plot(tipidcd,ipidcd,':ok','MarkerIndices',1:3:length(tipidcd),'Ma
rkerSize',3);
hold off;
xlim([tipidcd(1) tipidcd(end)]);
legend('Analog PID','Digital PID');
title({'Impulse Disturbance Response of Cart Position','under PID
Control'});
set(gcf,'color','w');

%% 4.) Display
%% 4.) - Analog and Digital PI controller
display(cpitf); %analog PI
display(gczpi); %digital PI

%% 4.) - Analog and Digital PID controller
display(cpidtf); %analog PID
display(gczpid); %digital PID

```

### **Inverted\_Pendulum\_LQR\_design.m**

```

%-----
%   Inverted_Pendulum_LQR_desing.m
%-----
%%      Description
%
%   This is a script to compute the model based design LQR of the
%   inverted pendulum on the cart, in discrete and continuous
%   time.
%
%%      Output
%
% Parameters:
%   - controllability_ct: controllability continuous time
%   inverted pendulumsystem
%   - controllability_dt: controllability discrete time inverted

```

```

% pendulum system
% - K: matrix k for continuous time system (pendulum on the
% cart)
% - Kd: matrix k for discrete time system (pendulum on the
% cart)
%
% Figures:
% -Impulse disturbance response under LQR control
% -Impulse disturbance response under discrete LQR control
%
%% Version
%
% Author: Felix Ruiz Torrez (torrezfelix@gmail.com)
% Creation date: april-2019
% Matlab version: R2018a
%
%% Revision
%
% V1.0 | 5-jan-2019 | Felix Ruiz Torrez | Creation

%% Program
run('Cart_Pendulum_identification.m');
close all;
clc;

Tst=0.03; %Implementation transmission sample time
MotorDeadZone=35; %Implementation motor dead zone

%% 1.) Definition
%% 1.) - Controllability definition
Cm=ctrb(sspendup);
controllability_ct = rank(Cm);
Cmd=ctrb(sspendupd);
controllability_dt = rank(Cmd);

%% 1.) - Matrix Q and R definition
%Q = [1,0,0,0;0,1,0,0;0,0,1,0;0,0,0,1];
%Q = [1,0,0,0;0,1,0,0;0,0,1e2,0;0,0,0,1e3];
Q = [1e2,0,0,0;0,1,0,0;0,0,1e3,0;0,0,0,1];
R = 1e4;

%% 2.) Computing
%% 2.) - Continuous LQR for the inverted pendulum on the cart
K = lqr(sspendup.a,sspendup.b,Q,R); %K matrix continuous
inverted pendulum

%% 2.) - Discrete LQR for the inverted pendulum on the cart
Kd = dlqr(sspendupd.a,sspendupd.b,Q,R); %K matrix discrete
inverted pendulum

%% 2.) - Close loop state space inverted pendulum
Ac = sspendup.a-sspendup.b*K;

```

```

Bc = sspendup.b;
Cc = [1,0,0,0;0,0,1,0];
Dc = [0;0];
states = {'x' 'x_dot' 'phi' 'phi_dot'};
inputs = {'u'};
outputs = {'x'; 'theta'};
sys_cl =
ss(Ac,Bc,Cc,Dc,'statename',states,'inputname',inputs,'outputname',
,outputs);

%% 2.) - Close loop discrete state space inverted pendulum
Acd = sspendupd.a-sspendupd.b*Kd;
Bcd = sspendupd.b;
sys_cld =
ss(Acd,Bcd,Cc,Dc,Ts,'statename',states,'inputname',inputs,'output
name',outputs);

%% 3.) Plotting
%% 3.) - Impulse Response Continuous LQR graph
figure('Name','Pendulum Impulse Response','NumberTitle','off');
impz(sys_cl,'k');
title('Impulse Disturbance Response under LQR');
set(gcf,'color','w'); %set background color to white

%% 3.) - Impulse Response Discrete LQR graph
figure('Name','Pendulum Impulse Response','NumberTitle','off');
impz(sys_cld,'k');
title('Impulse Disturbance Response under Discrete LQR');
set(gcf,'color','w'); %set background color to white

%% 4.) Display
%% 4.) - Controllability for the inverted pendulum on the cart
%system
display(controllability_ct); %continuous time
display(controllability_dt); %discrete time

%% 4.) - Optimal gains for the inverted pendulum on the cart
%system
display(K); %continuous time
display(Kd); %discrete time

```



## **Anexo 7. Manual de usuario**

En este manual se describen las funciones, características, conexiones y correcto uso del péndulo invertido que será instalado en el Laboratorio de Automatización de la Facultad de Electrotecnia y Computación de la UNI. El objetivo es informar sobre el procedimiento adecuado para su utilización, así como los principales cuidados que se deben de tener para su funcionamiento.

El péndulo invertido es un sistema físico utilizado como banco de pruebas para Sistemas de Control. Su problema radica en diseñar un controlador que mantenga la varilla, a través del desplazamiento del carro, de manera vertical sin que caiga hacia los lados.

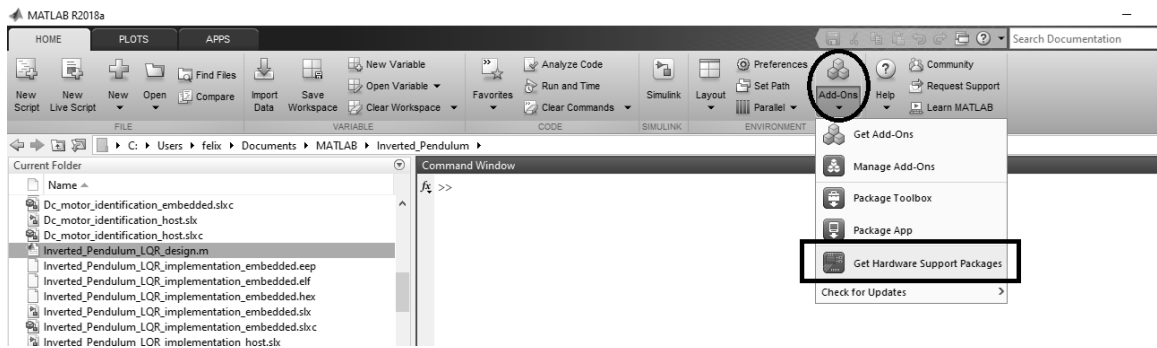
### **Utilidad**

El péndulo invertido es un sistema que sirve como herramienta de estudio ya que permite al usuario relacionar la teoría de Sistemas de Control con su implementación, el cual puede manipularlo y observar distintos resultados al variar los parámetros. Para este péndulo invertido se ha diseñado tanto un compensador PID como un compensador LQR.

El sistema del péndulo invertido sobre el carro cuenta de dos subsistemas que trabajan acoplados, los cuales son el carro y la varilla. Ambos cuentan con encoders o codificadores de cuadratura, que permiten la medición de las posiciones tanto traslacional del carro, como rotacional del péndulo, las cuales son realimentadas para la estabilización del sistema.

### **Instalación de Matlab, toolbox de Arduino y Rensselaer**

1. Instalar Matlab R2018a.
2. Instalar las toolbox de simulink para trabajar con Arduino (seguir los siguientes pasos):
  - Ir a la pestaña HOME, Add-Ons y dar click en Get hardware Support Package.



- Buscar Simulink Support Package for Arduino Hardware e instalar.



- Ingresar con su cuenta de Mathworks o crear una nueva.

×

### Sign in to your MathWorks Account

**Email Address or User ID:**

**Password:**

Forgot Password?

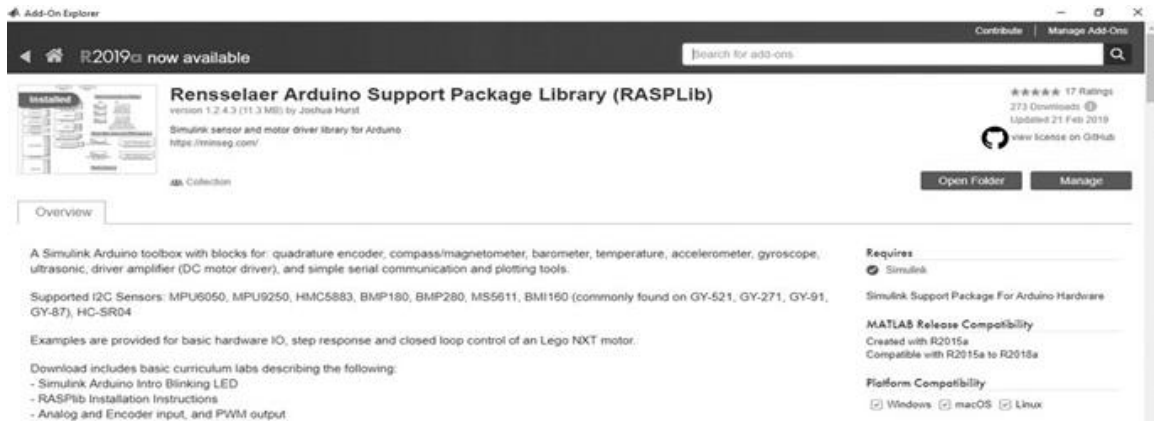
☒ Keep me signed in

*You are logging in from within MATLAB to access unsupported MathWorks and third-party authored content made available under separate license terms.*

Sign In

Create Account

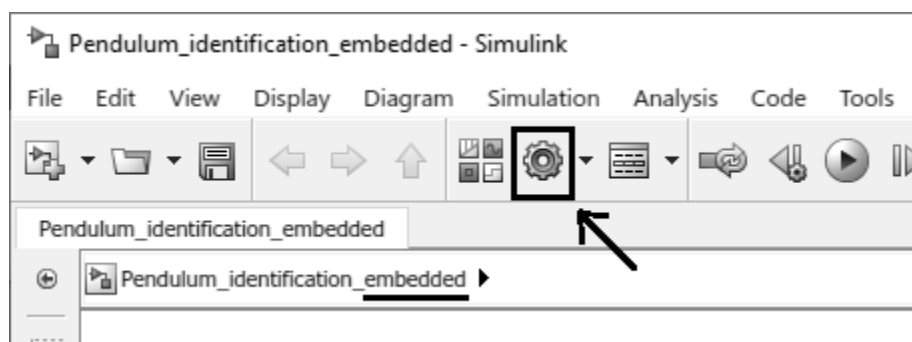
- Buscar Rensselaer Arduino support package Library e instalar.



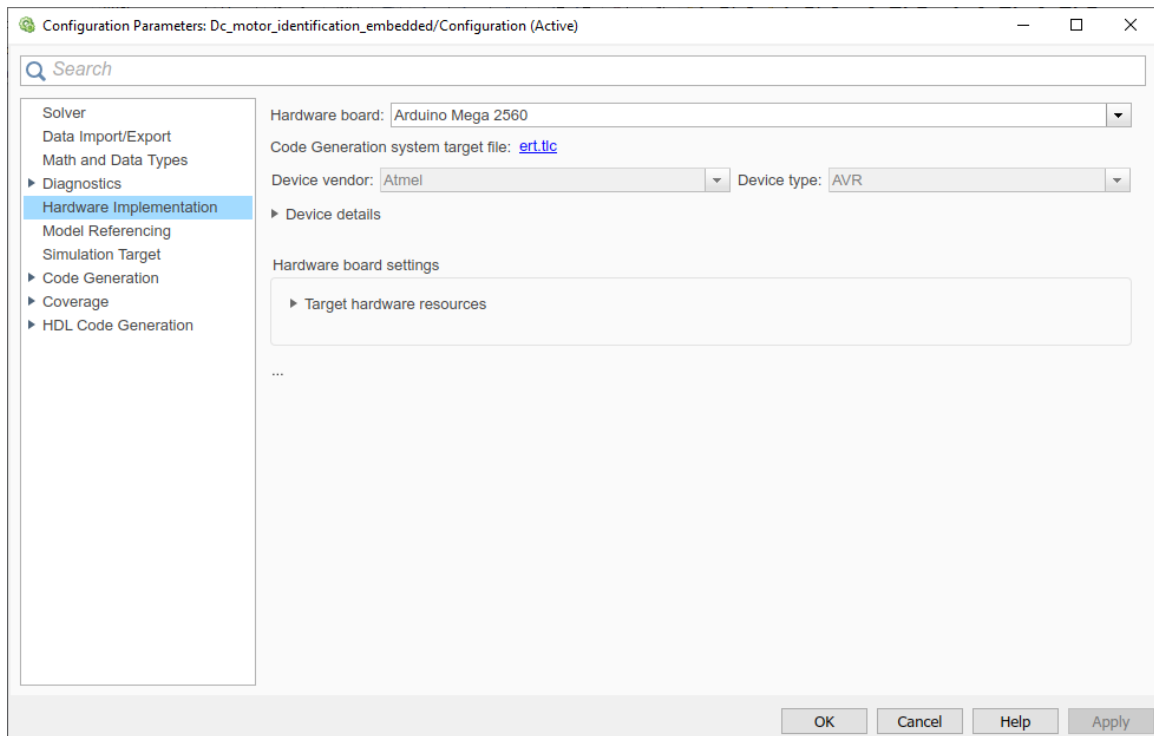
## Configuración de los programas de Simulink para trabajar con Arduino

Los programas de Simulink que se queman en el Arduino Mega 2560 vienen con la palabra embedded en su nombre y las configuraciones se muestran a continuación para su verificación.

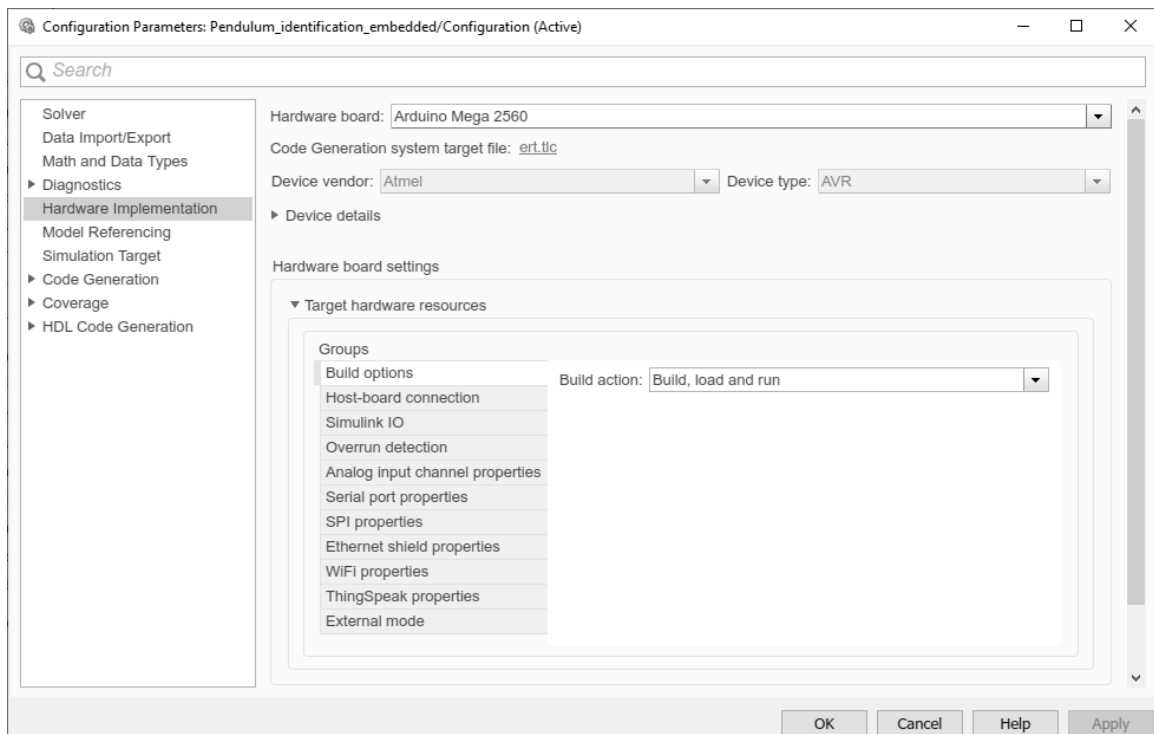
- Dar click en el botón model configuration parameters



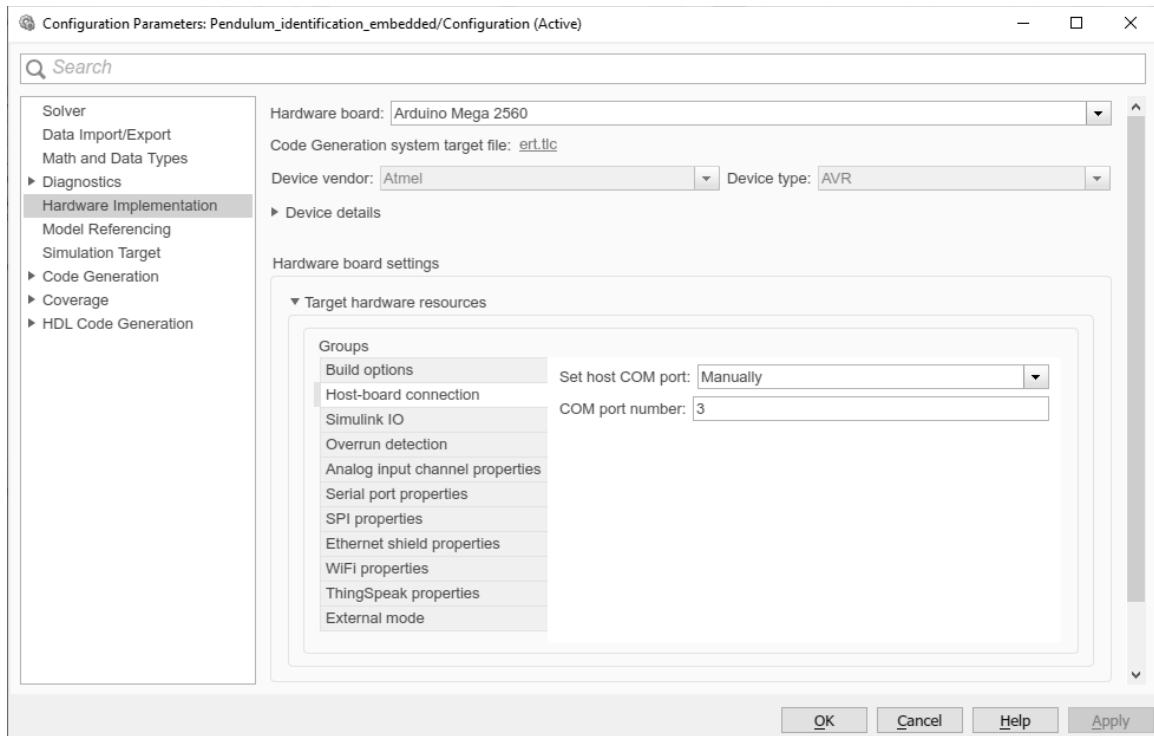
- Dar click en Hardware Implementation y validar que esté seleccionado el Hardware board Arduino Mega 2560.



- Expandimos Target hardware resources y en la pestaña Build options debe estar seleccionado Build, load and run.

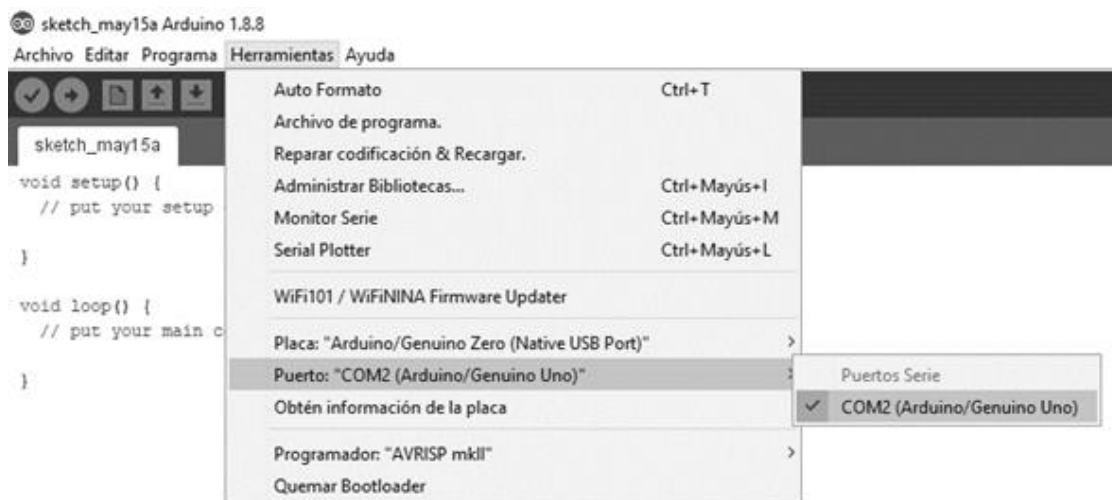


- En la pestaña Host-board connection establecemos que la conexión sea manual y configuramos el puerto en el que se encuentra conectada la tarjeta Arduino.

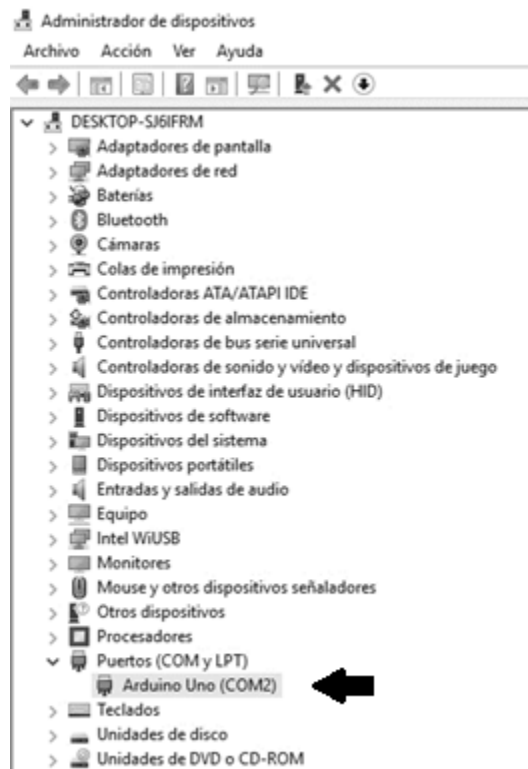


Para saber en cual puerto se encuentra conectada la tarjeta Arduino podemos hacerlo de dos maneras:

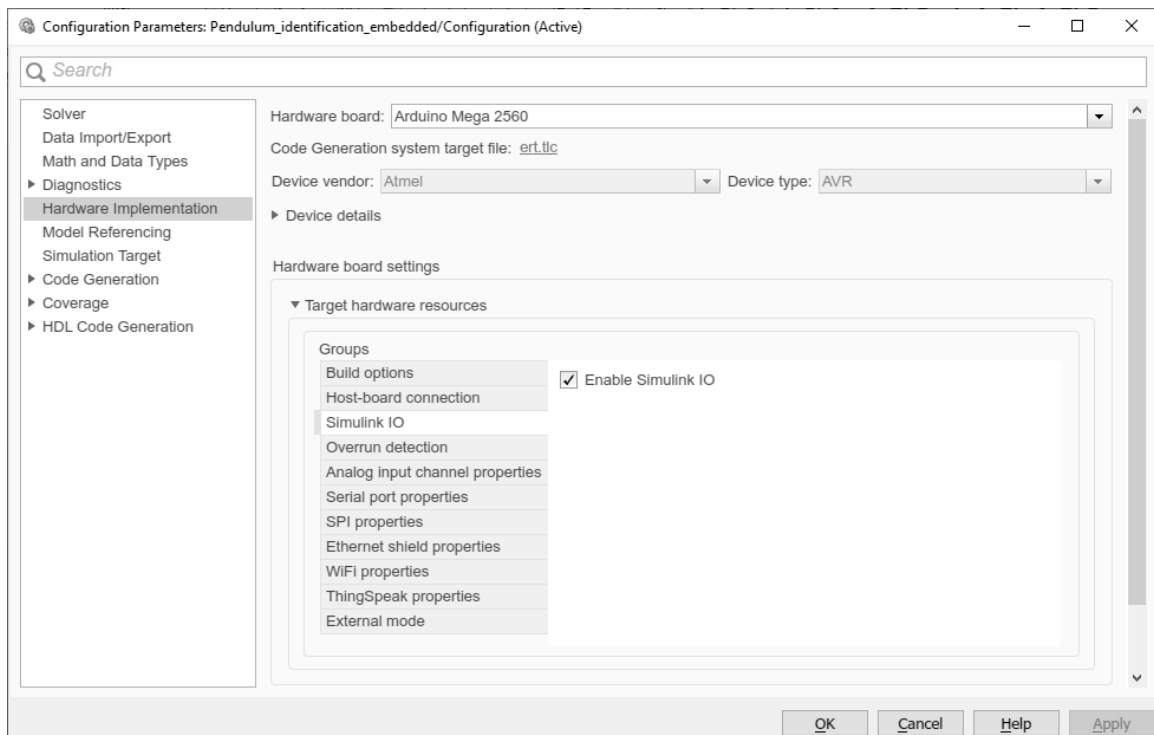
#### Opción 1: Haciendo uso de la plataforma Arduino IDE



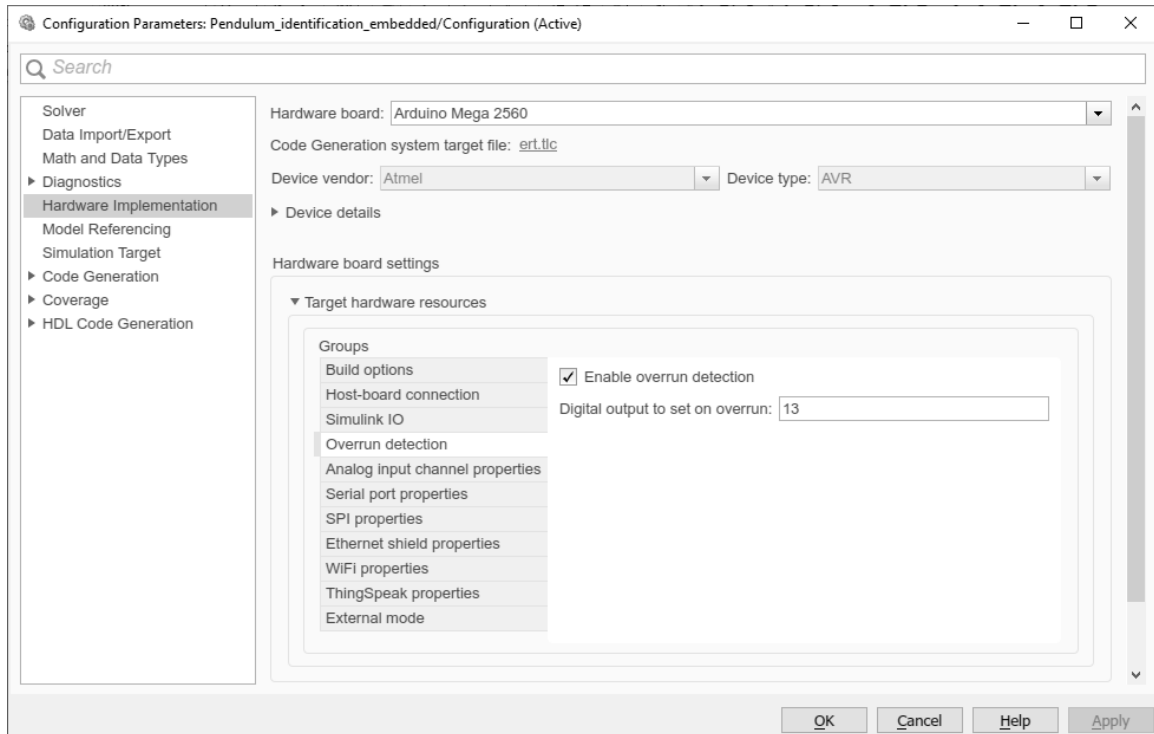
## Opción 2: Desde el administrador de dispositivos



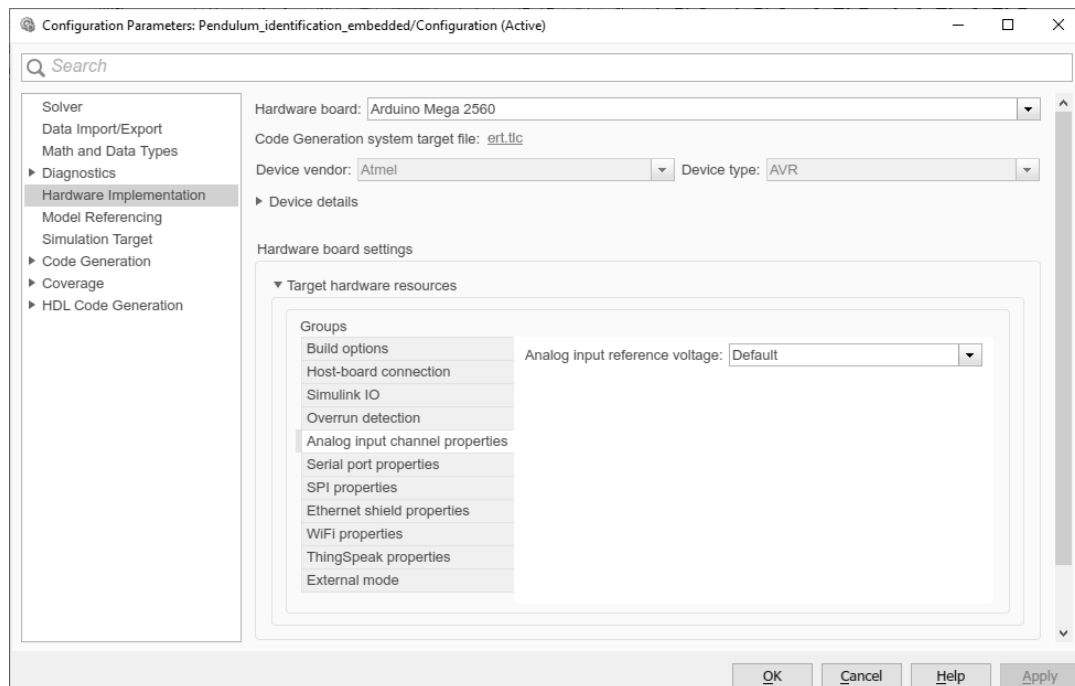
- En la pestaña Simulink IO habilitamos la casilla.



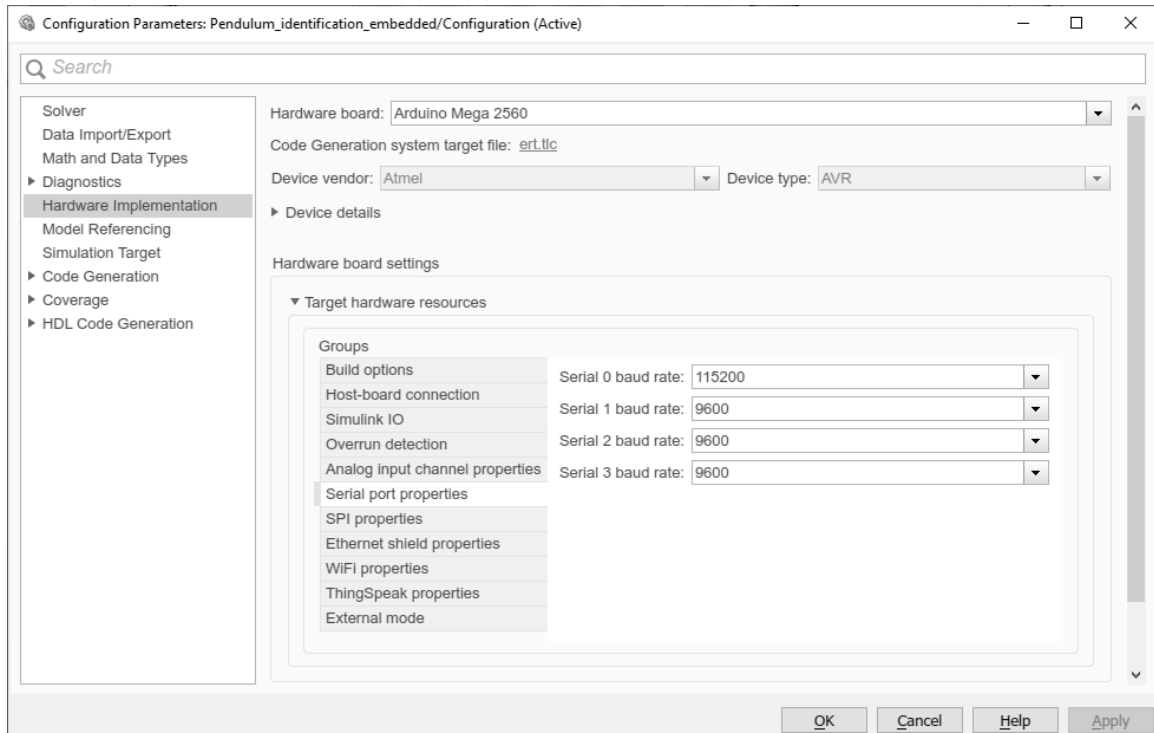
- En la pestaña Overrun detection establecemos que se active el pin 13 de la tarjeta Arduino. Si el programa embebido en el Arduino no se está ejecutando en tiempo real, el led conectado al pin 13 se enciende.



- En la pestaña Analog Input channel properties queda Default.



- En la pestaña Serial port properties cambiamos serial 0 baud rate a 115200. Esta es la conexión usb del Arduino al computador.



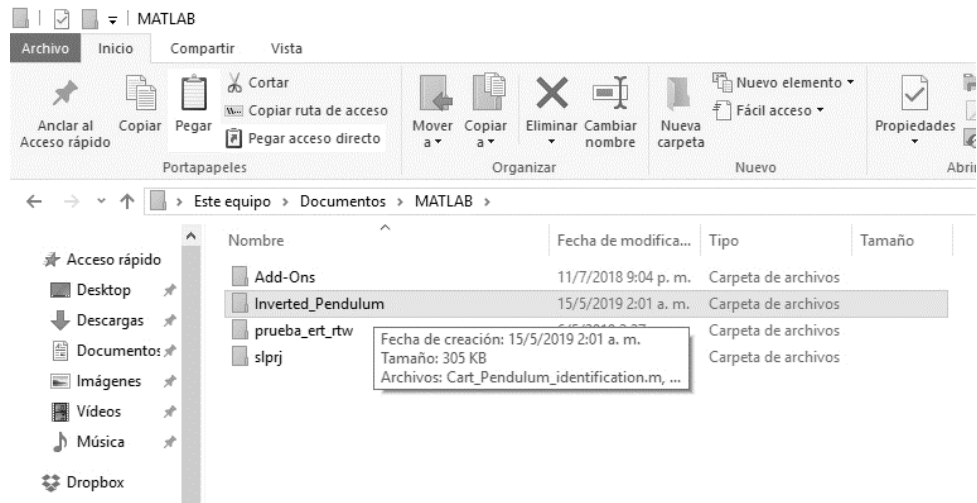
- Las configuraciones restantes no se modifican.

## Pasos para seguir

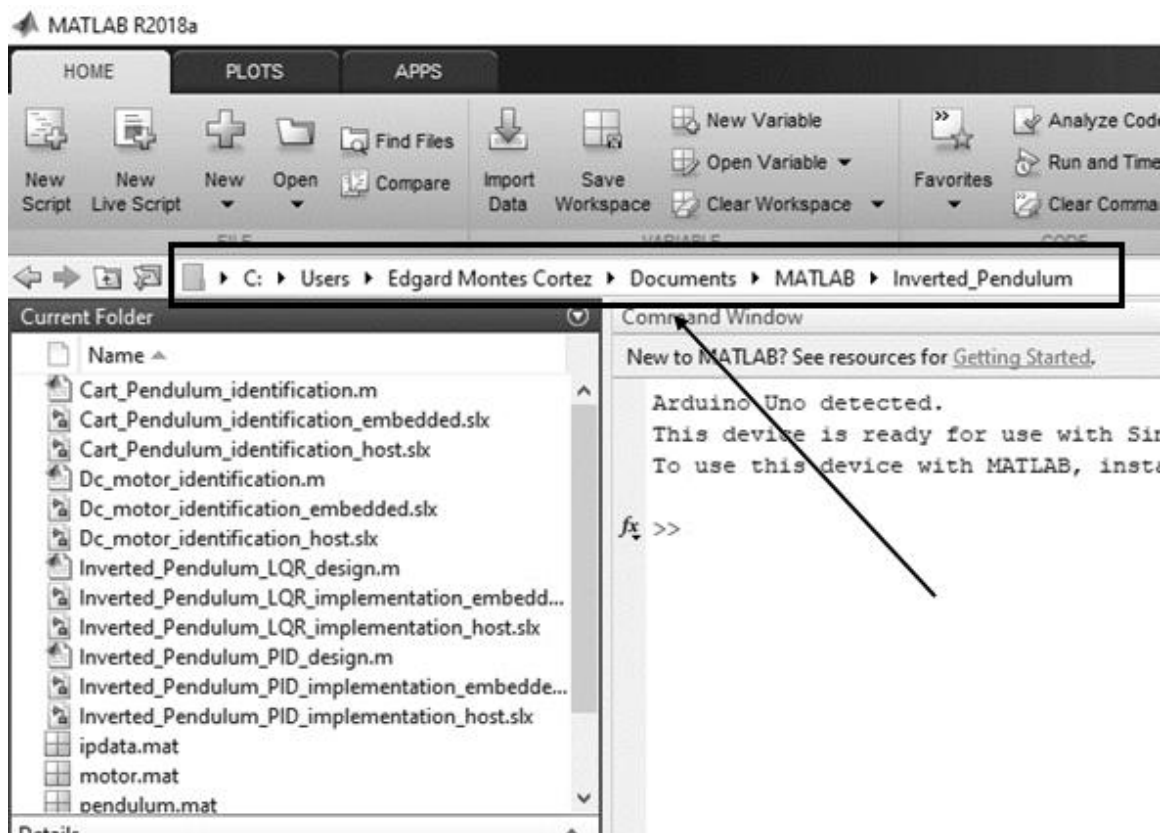
- Colocar la maqueta del péndulo en una mesa o superficie plana nivelada.
- Conectar el adaptador de 12 VDC a la corriente alterna (110 VAC) y al conector/Jack DC hembra de 2.1 mm ubicado en la parte posterior del marco del prototipo.
- Conectar la tarjeta Arduino mediante el cable usb al computador.
- Copiar la carpeta Inverted\_Pendulum en el directorio C:\Users\Documents\MATLAB. Esta carpeta contiene todos los archivos a utilizar y se pueden obtener del dvd que acompaña a este documento o se pueden descargar del siguiente link:

[https://drive.google.com/open?id=1X\\_FtC-ixHem8\\_9K\\_LI3eqbgxQeoiO4d3](https://drive.google.com/open?id=1X_FtC-ixHem8_9K_LI3eqbgxQeoiO4d3)





- Abrir Matlab y ubicarnos en la ruta donde copiamos carpeta.

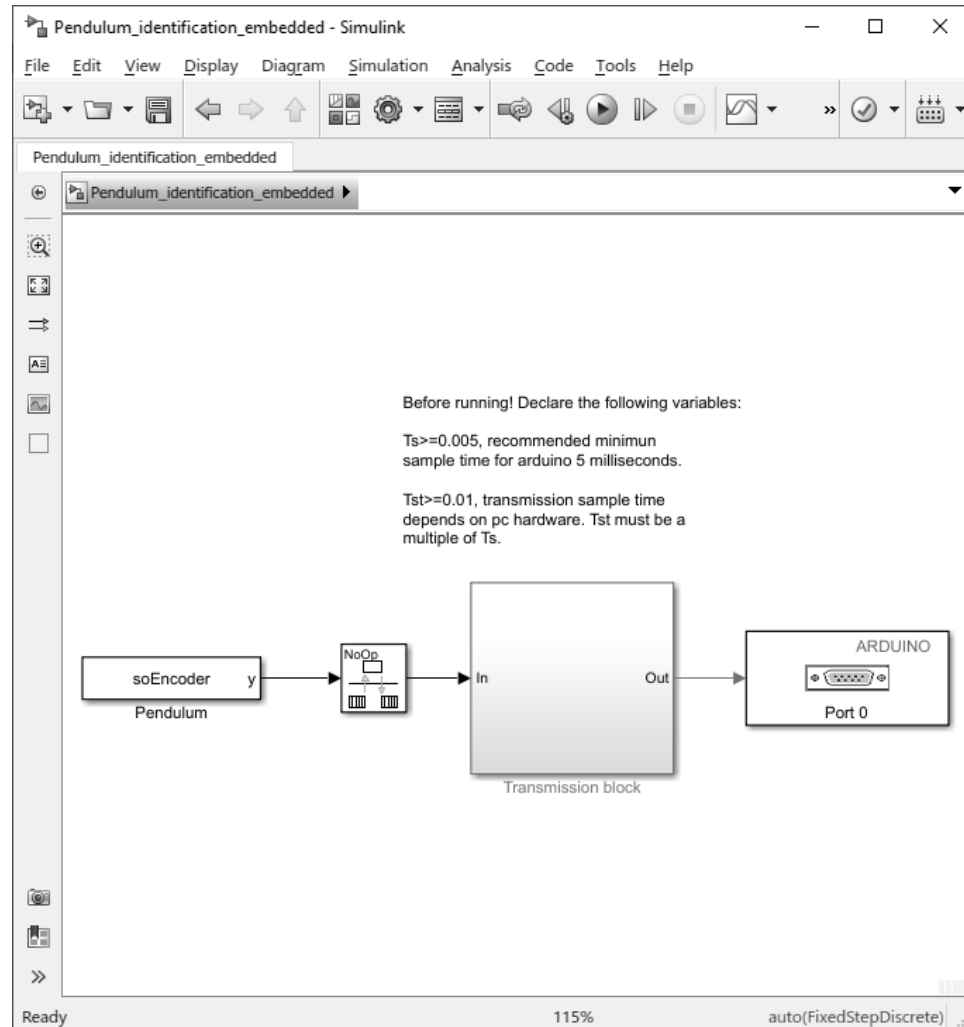


## Identificación de Parámetros del Péndulo

### Adquisición de datos:

- Atornillar la varilla al eje del encoder sobre el carro quedando hacia abajo.

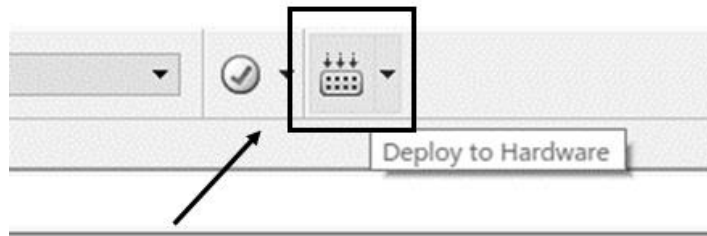
- Abrir el programa de Simulink *Pendulum\_identification\_embedded.slx*



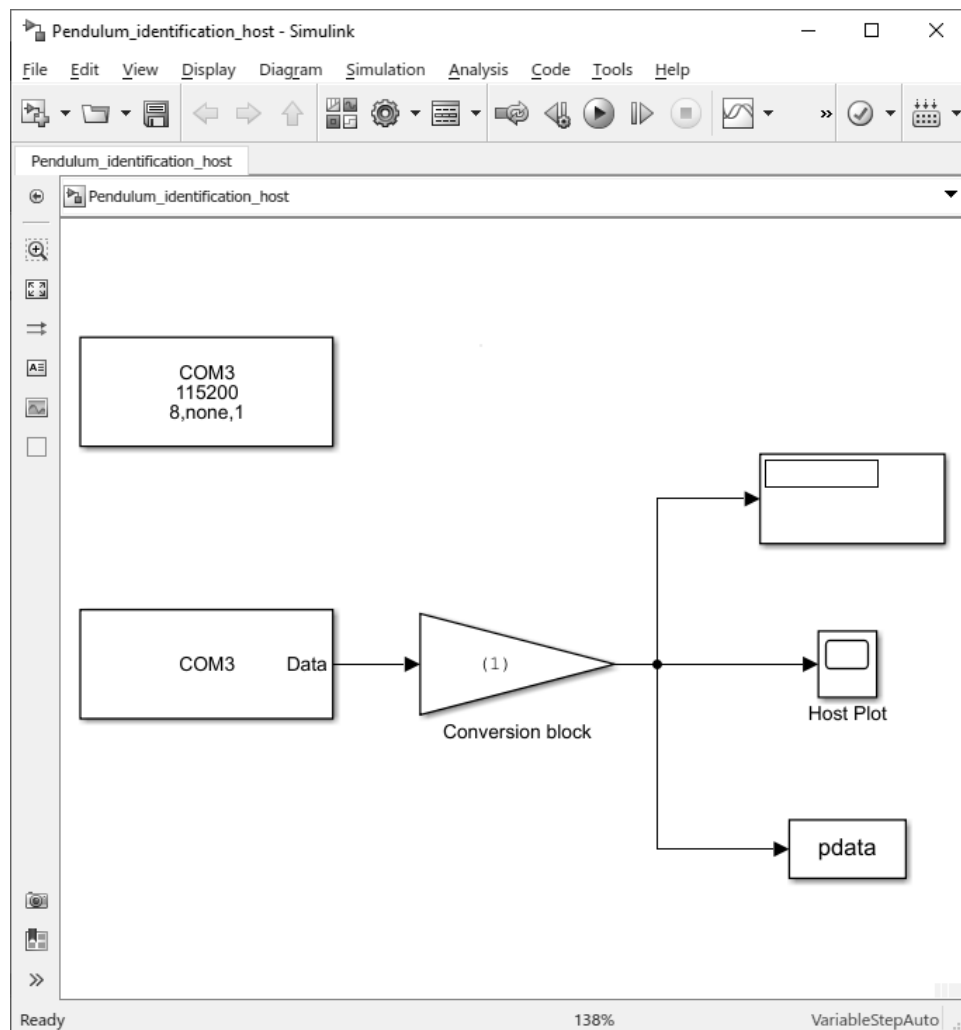
- Declarar en el Matlab prompt el tiempo de muestreo de todo el sistema  $T_s$  mayor o igual a 5 milisegundos, y el tiempo de muestreo  $T_{st}$  de transmisión de datos mayor o igual a 10 milisegundos.  $T_s$  tiene que ser múltiplo de  $T_{st}$ . El tiempo de la transmisión de datos va a variar en dependencia de la capacidad del computador de correr en tiempo real el programa de Simulink *Pendulum\_identification\_host.slx* que se ejecuta como host en el computador.

```
>> Ts=0.005;
>> Tst= 0.01;
```

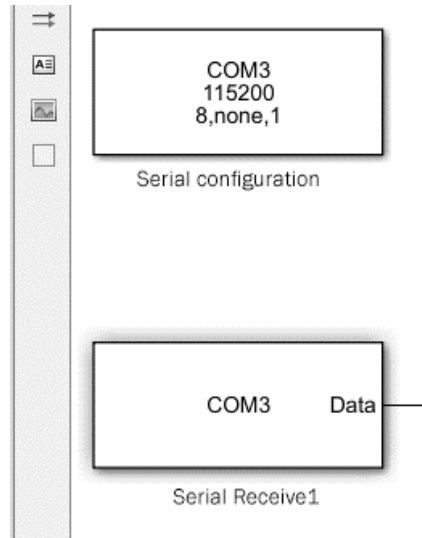
- Quemar el programa de Simulink *Pendulum\_identification\_embedded.slx* en el Arduino.



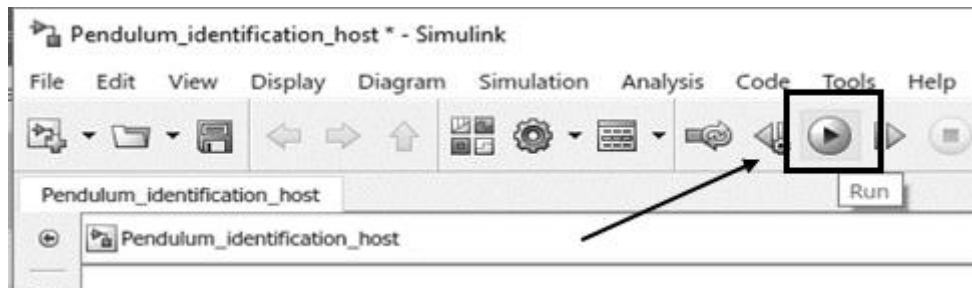
- Abrir el programa de Simulink *Pendulum\_identification\_host.slx*. Éste en conjunto con el programa *Pendulum\_identification\_embedded.slx* son necesarios para la adquisición de datos.



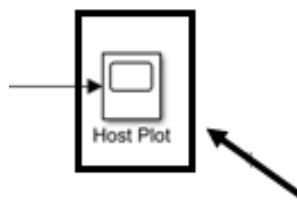
- Verificar que en los bloques Serial Receive y Serial configuration esté seleccionado el puerto COM del Arduino.

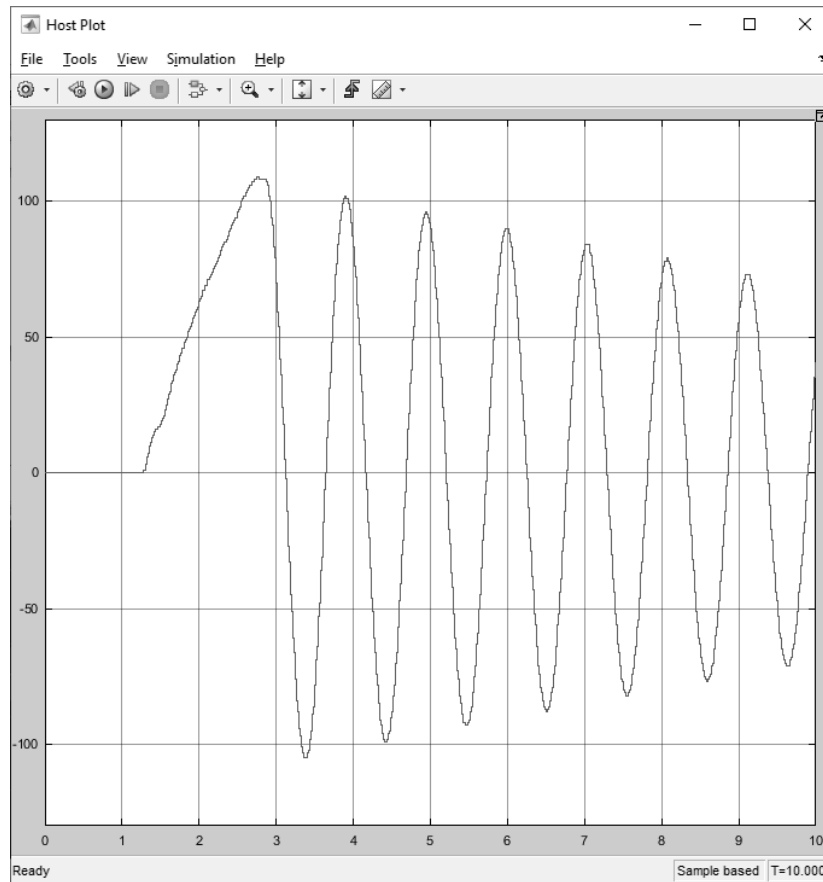


- Dar click en el botón Run.



- Dar doble click en el icono Host Plot y soltar el péndulo desde un ángulo de 10 grados o su equivalente a 113 pulsos del encoder. Se observará la gráfica del ángulo del péndulo en tiempo real.





- Estos datos se guardan en la variable pdata en el Workspace de Matlab para su posterior análisis.

#### Identificación del modelo:

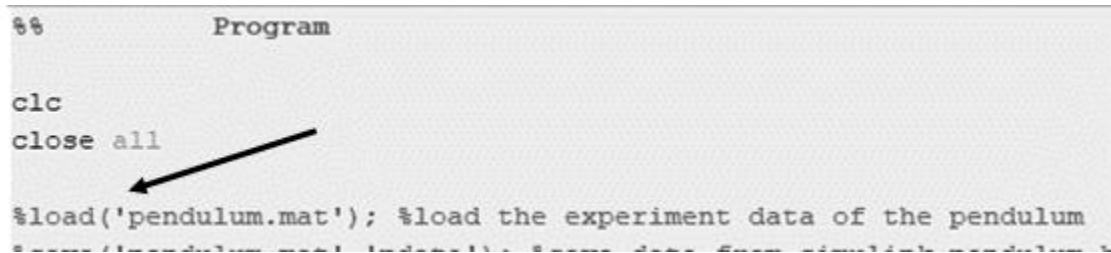
- Abrir el script de Matlab Pendulum\_identification.m
- La carpeta Inverted\_pendulum contiene datos experimentales que hemos brindado con anticipación, que si se desean cargar, debe quedar sin comentar la línea de código `load('pendulum.mat');`

```
%%      Program

clc
close all

load('pendulum.mat'); %load the experiment data of the pendulum
%load('pendulum.mat'); %load the experiment data of the pendulum
```

- Si se desea utilizar los datos del workspace de Matlab, comentar la línea de código `load('pendulum.mat');`



```
%%
Program

clc
close all

%load('pendulum.mat'); %load the experiment data of the pendulum
```

- En la pestaña Editor, dar click en Run o presionar F5 para ejecutar el script *Pendulum\_identification.m*



- El script da como resultado las figuras 33, 34 y 35 del acápite 2.3. En la ventana de comando se observan la ecuación diferencial del péndulo y su solución.

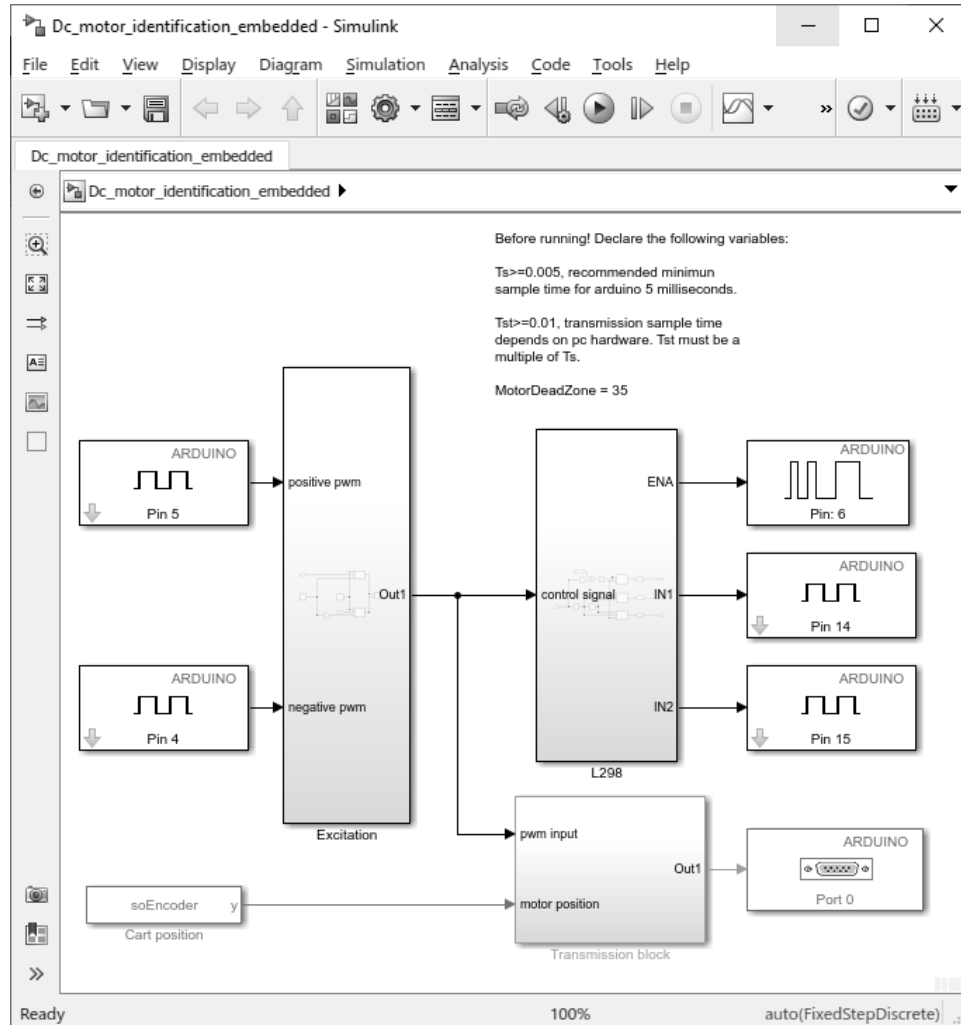
## Identificación de los parámetros del carro

### Adquisición de datos:

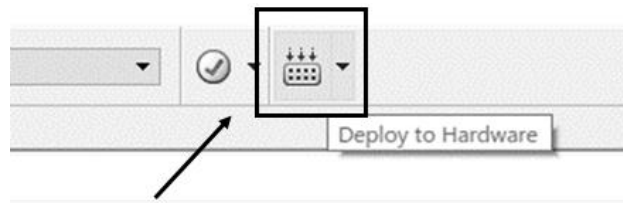
- Quitar la varilla del eje del encoder sobre el carro.
- Declarar en el Matlab prompt el tiempo de muestreo de todo el sistema  $T_s$  mayor o igual a 5 milisegundos, el tiempo de muestreo  $T_{st}$  de transmisión de datos mayor o igual a 10 milisegundos y la zona muerta del motor  $MotorDeadZone$  a 35.  $T_s$  tiene que ser múltiplo de  $T_{st}$ . El tiempo de la transmisión de datos va a variar en dependencia de la capacidad del computador de correr en tiempo real el programa de Simulink *Dc\_motor\_identification\_host.slx*.

```
>> Ts=0.005;
>> Tst= 0.01;
>> MotorDeadZone=35;
```

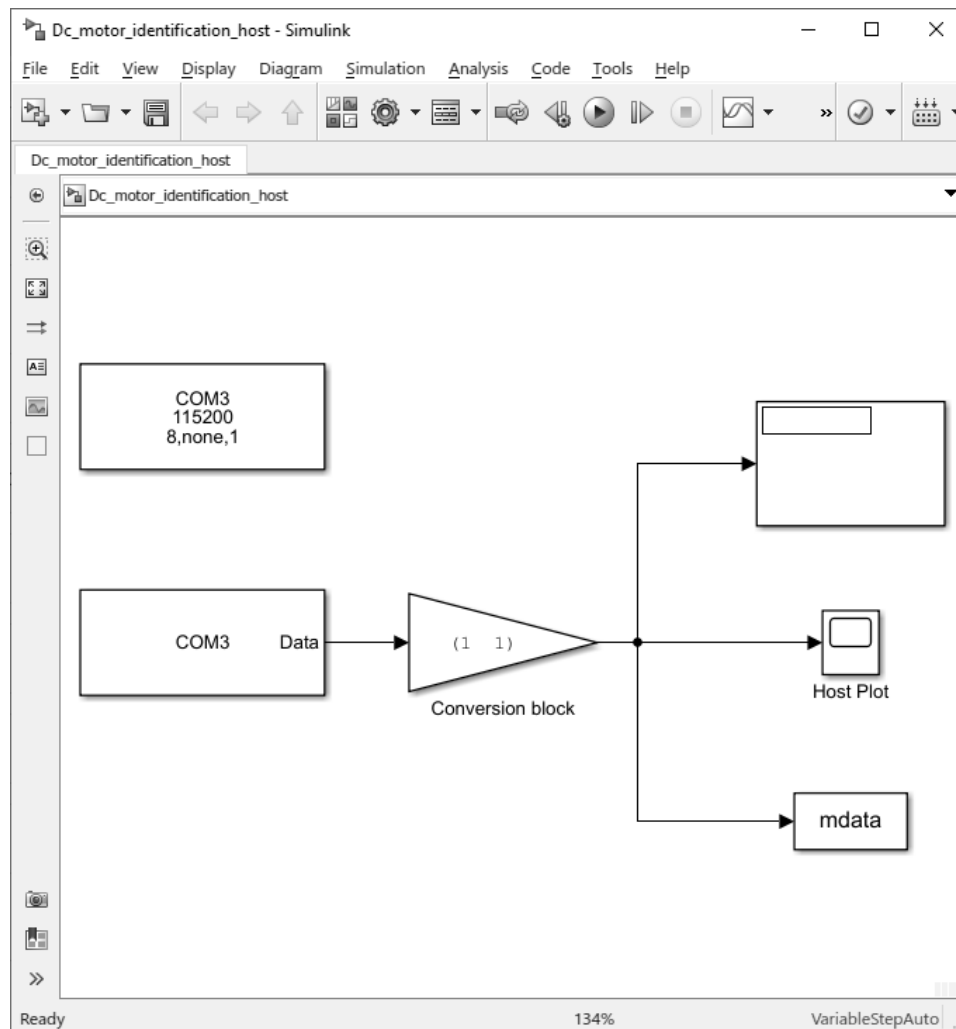
- Abrir el programa de Simulink *Dc\_motor\_identification\_embedded.slx*



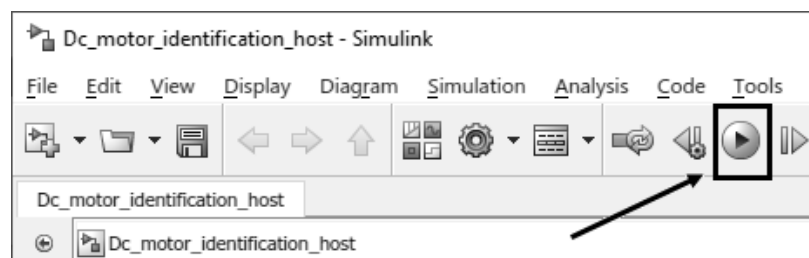
- Quemar el programa de Simulink *Dc\_motor\_identification\_embedded.slx* en el Arduino.



- Abrir el programa *Dc\_motor\_identification\_host.slx*

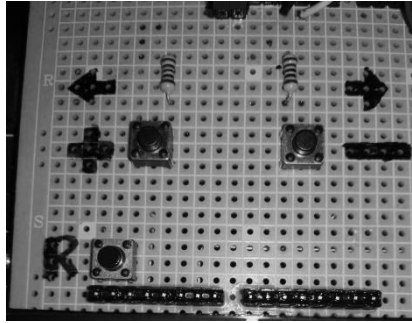


- Verificar que en los bloques Serial Receive y Serial configuration esté seleccionado el puerto COM del Arduino.
- Dar click en el botón Run.

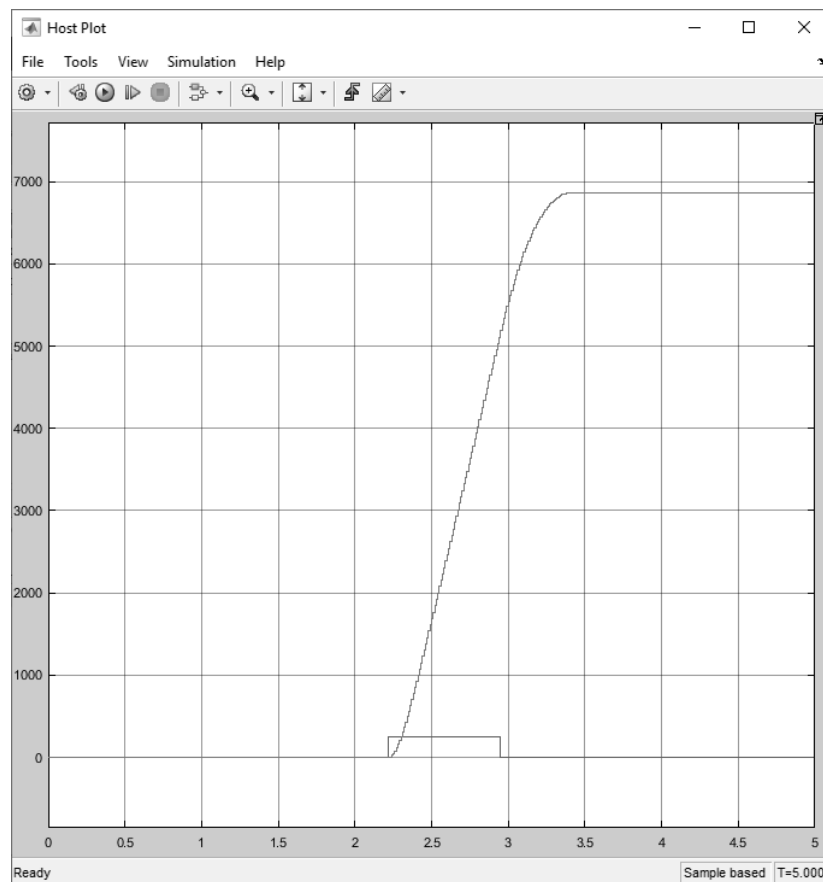


- Presionar el botón + o – en la tarjeta para mover el carro de un extremo a otro y crear una excitación pulso.





- Dar doble click en el icono Host Plot para observar la gráfica de la excitación y la posición del carro en tiempo real.



- Estos datos se guardan en la variable mdata en el Workspace de Matlab para su posterior análisis.

#### Identificación del modelo:

- Abrir el script de Matlab *Dc\_motor\_identification.m*.

- La carpeta Inverted\_Pendulum contiene datos experimentales que hemos brindado con anterioridad, que si se desean cargar debe quedar sin comentar la línea de código `load('motor.mat');`

```
%%      Program
clc;
%clear all;
close all;

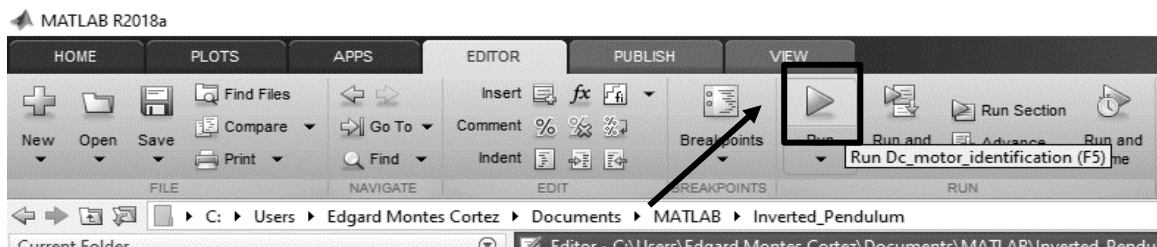
load('motor.mat');
%save('motor.mat','mdata');
```

- Si se desea utilizar los datos del workspace de Matlab, comentar la línea de código `load('motor.mat');`

```
%%      Program
clc;
%clear all;
close all;

%load('motor.mat');
%save('motor.mat','mdata');
```

- Si se desea guardar los datos del Workspace, no comentar la línea de código `save('motor.mat','mdata');`
- En la pestaña Editor, dar click en Run o presionar F5 para ejecutar el script *Dc\_motor\_identification.m*

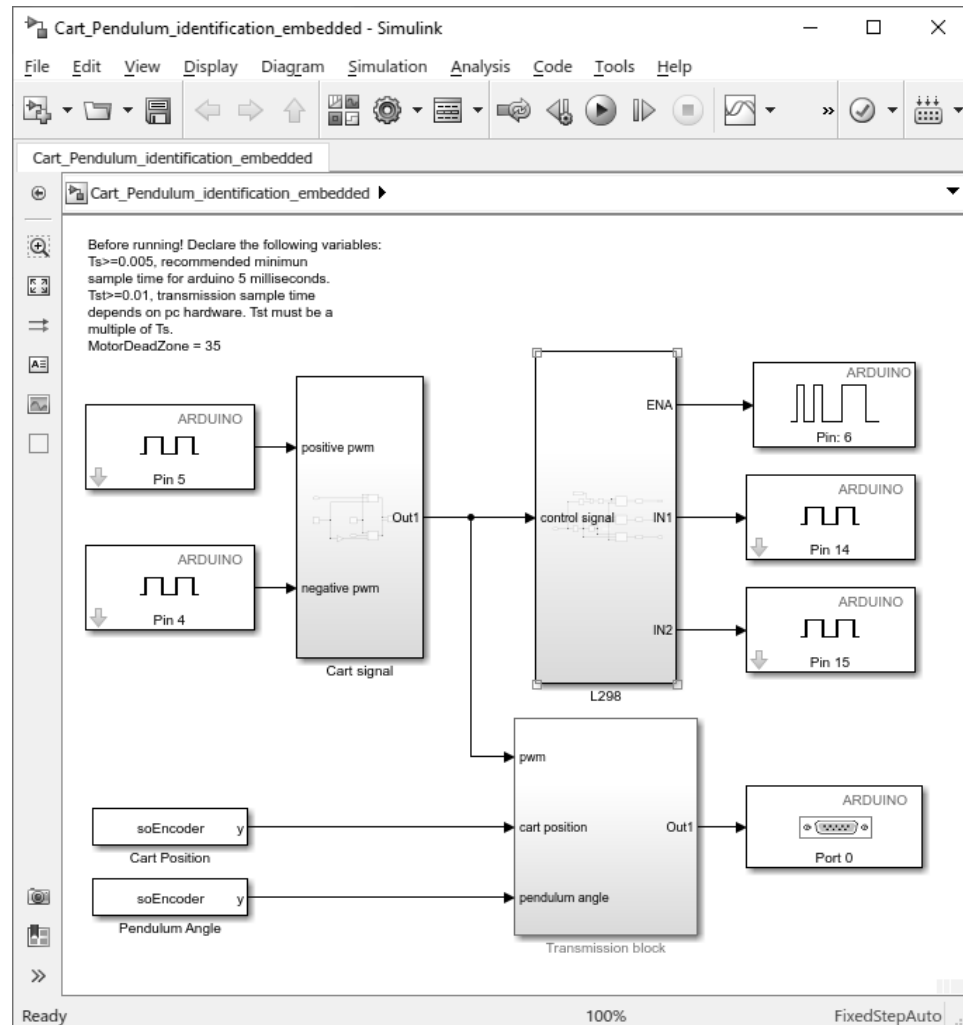


- El script da como resultado las figuras 36, 37 y 38 del acápite 2.4. En la ventana de comandos se observan la ganancia del motor  $K_m$ , su constante de tiempo  $T_m$  y su función de transferencia *motor\_tf*.

## Identificación de los parámetros del péndulo sobre el carro

### Adquisición de datos:

- Atornillar la varilla al eje del encoder sobre el carro quedando hacia abajo.
- Abrir el programa de Simulink *Cart\_Pendulum\_identification\_embedded.slx*.

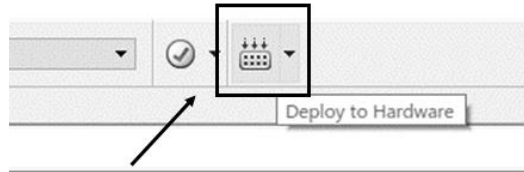


- Declarar en el Matlab prompt el tiempo de muestreo de todo el sistema Ts mayor o igual a 5 milisegundos, el tiempo de muestreo Tst de transmisión de datos mayor o igual a 10 milisegundos y la zona muerta del motor MotorDeadZone igual a 35. Ts tiene que ser múltiplo de Tst. El tiempo de la transmisión de datos va a variar en dependencia de la capacidad del computador de correr en tiempo real el programa de Simulink

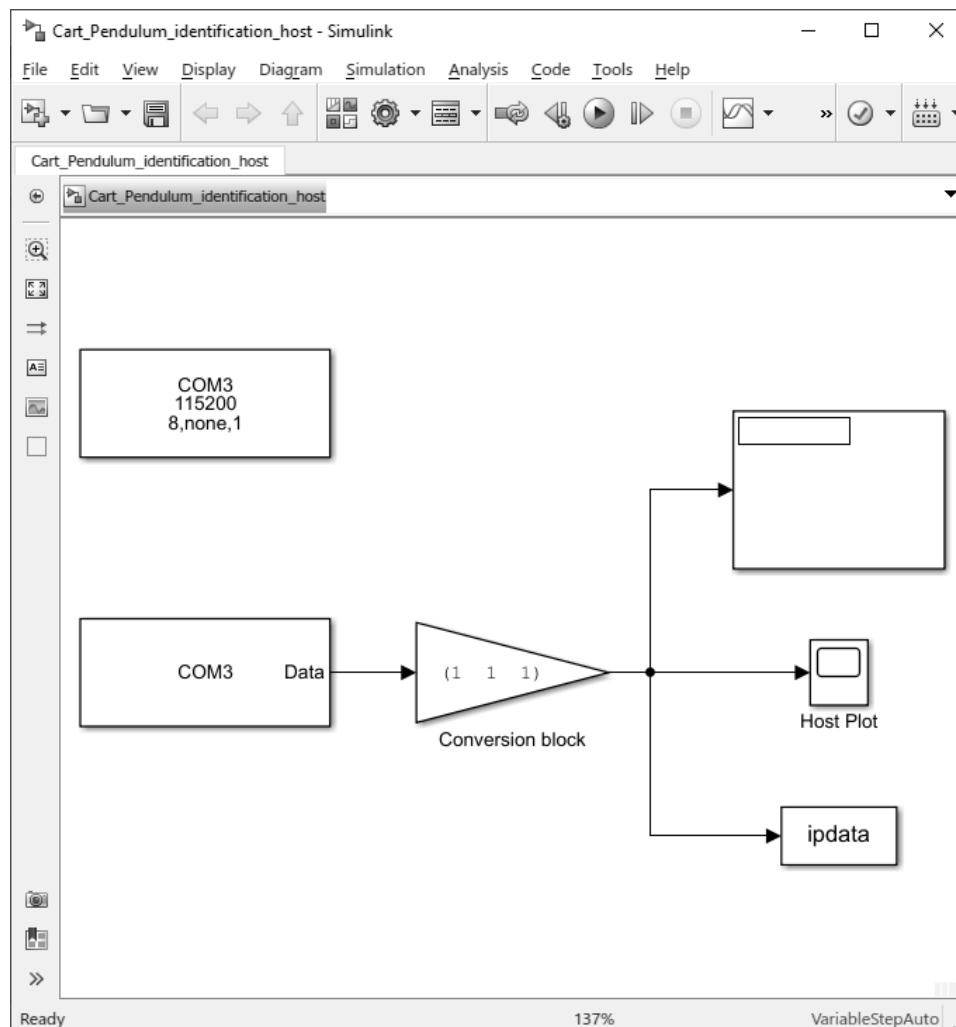
*Dc\_motor\_identification\_host.slx* que se ejecuta como host en el computador.

```
>> Ts=0.005;  
>> Tst= 0.01;  
>> MotorDeadZone=35;
```

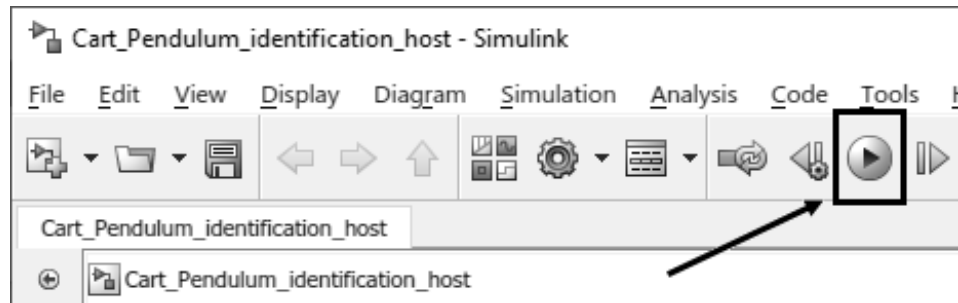
- Quemar el programa de simulink *Cart\_Pendulum\_identification\_embedded.slx* en el Arduino.



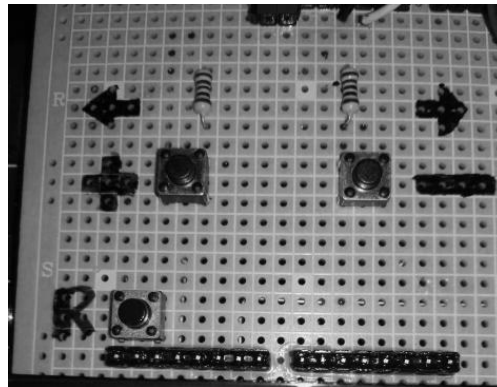
- Abrir el programa *Cart\_Pendulum\_identification\_host.slx* para la adquisición de datos.



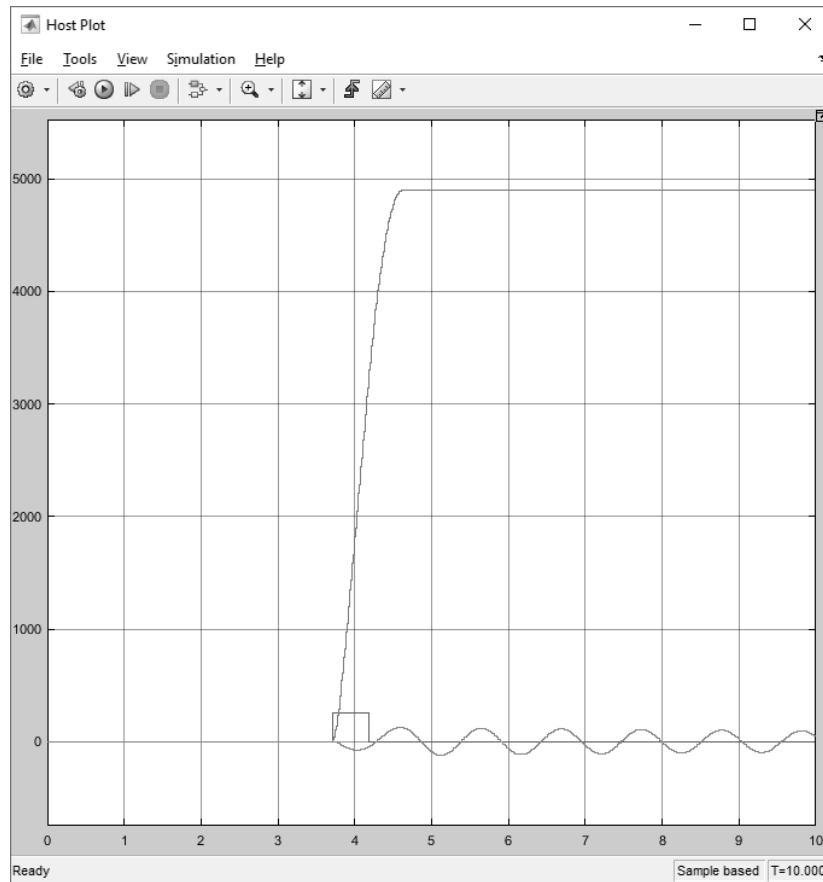
- Verificar que en los bloques Serial Receive y Serial configuration esté seleccionado el puerto COM del Arduino.
- Dar click en el botón Run.



- Presionar el botón + o – en la tarjeta para crear una excitación pulso que mueve el carro de un extremo a otro y simultáneamente que el péndulo oscile.



- Dar doble click en el icono Host Plot para observar las señales de excitación, la posición del carro y el ángulo del péndulo en tiempo real.



- Estos datos se guardan en la variable `ipdata` en el Workspace de Matlab para su posterior análisis.

#### Identificación del modelo:

- Abrir el script de Matlab *Cart\_Pendulum\_identification.m*.
- La carpeta *Inverted\_Pendulum* contiene datos experimentales que hemos brindado con anticipación, que si se desean cargar debe quedar sin comentar la línea de código `load('ipdata.mat');`

---

```

%%          Program
clc
close all

load("ipdata.mat") %comment this line if it is used data from workspace
%save('ipdata.mat','ipdata');
```

- Si los datos a utilizar ya están en el workspace de Matlab, comentar la línea de código `load('ipdata.mat');`

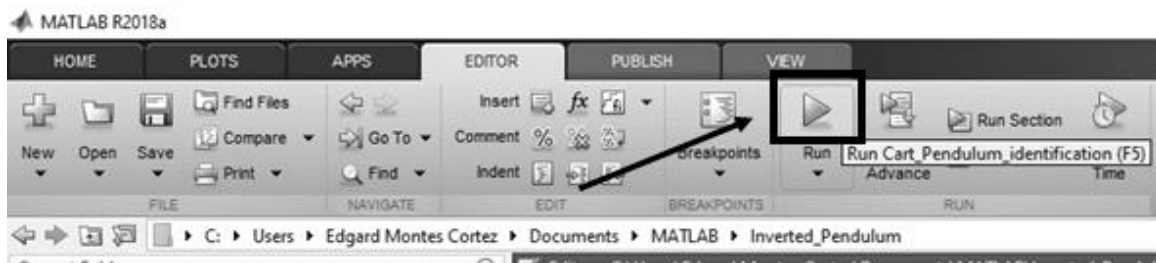
- Si se desea guardar los datos del Workspace, no comentar la línea de código `save('ipdata.mat','ipdata');`
- Modificar el tiempo de muestreo del modelo en tiempo discreto a conveniencia. Este modelo en tiempo discreto se utiliza posteriormente en el diseño e implementación del PID y LQR.

```

Cart_Pendulum_identification.m  X  +
37      %load("ipdata.mat")  %comment this l:
38      %save('ipdata.mat','ipdata');
39
40 -  Ts = 0.005;  %system sample time

```

- En la pestaña Editor, dar click en Run o presionar F5 para ejecutar el script *Cart\_Pendulum\_identification.m*




- El script da como resultado de la figura 39 a la 42, de los acápites 2.5 y 2.6. En la ventana de comando de Matlab se obtienen los modelos en espacio de estados del péndulo hacia abajo sobre el carro en tiempo continuo y discreto, del péndulo invertido sobre el carro en tiempo continuo y discreto, además de la función de transferencia del péndulo invertido.

## Diseño e implementación del controlador PID

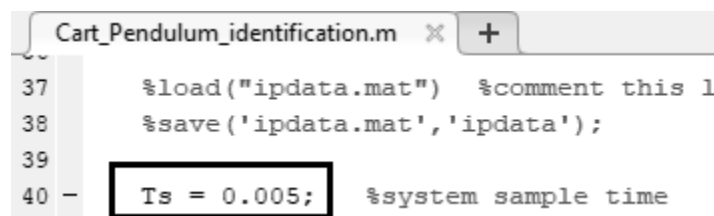
### Diseño:

- Abrir el script de Matlab *Inverted\_Pendulum\_PID\_design.m*. El diseño del PID está basado en el modelo del péndulo invertido sobre el carro, por lo que script llama a *Cart\_Pendulum\_identification.m* mediante la línea de comando: `run('Cart_Pendulum_identification.m');`
- El tiempo de muestreo  $T_s$  de transmisión de datos al igual que la zona muerta de motor están declarados en el script de Matlab *Inverted\_Pendulum\_PID\_design.m*. Estas variables se utilizan posteriormente en los programas de Simulink *Inverted\_Pendulum\_PID\_implementation\_embedded.m* e *Inverted\_Pendulum\_PID\_implementation\_host.m*. Modificar estas variables a conveniencia.



```
Inverted_Pendulum_PID_design.m  X  +
43
44 -   Tst=0.03;    %Implementation transmission sample time
45 -   MotorDeadZone=35;  %Implementation motor dead zone
46
```

- En el script *Cart\_Pendulum\_identification.m* modificar el tiempo de muestreo  $T_s$  del modelo discreto a utilizar en el diseño del PID. También  $T_s$  es el tiempo de muestreo de todo el sistema en el programa de Simulink *Inverted\_Pendulum\_PID\_implementation\_embedded.slx* que va embebido en el Arduino Mega 2560. De igual manera, comentar la línea de código `load('ipdata.mat');` si los datos a utilizar yacen del Workspace de Matlab.



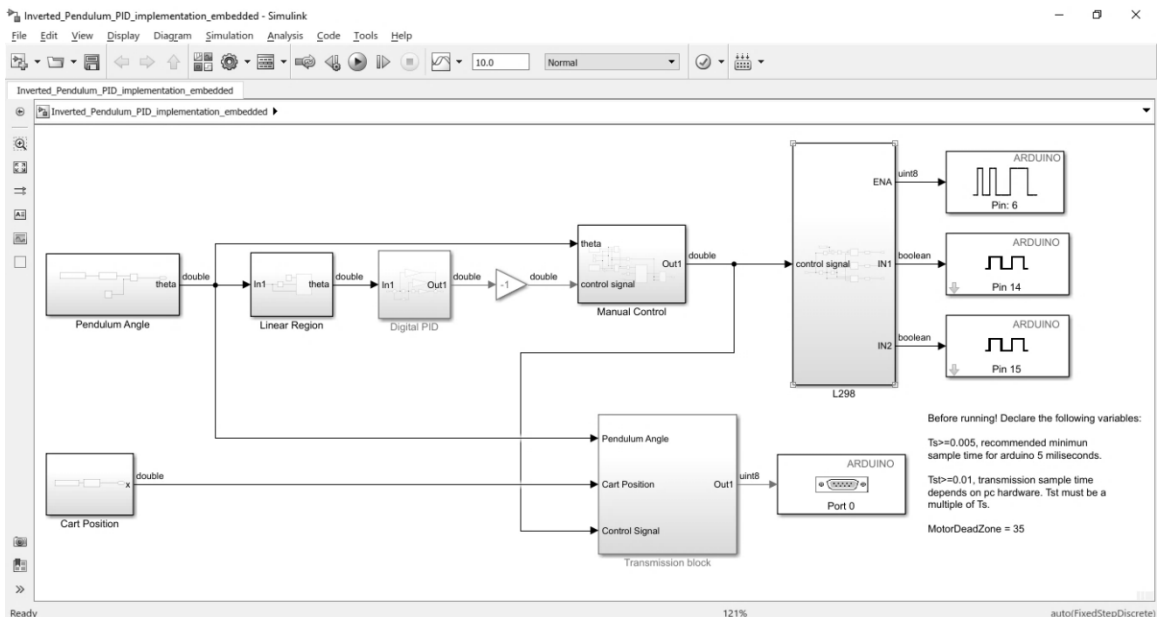
```
Cart_Pendulum_identification.m  X  +
37   %load("ipdata.mat") %comment this l:
38   %save('ipdata.mat','ipdata');
39
40 -   Ts = 0.005;    %system sample time
```



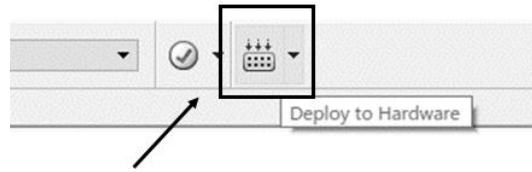
- 
- A screenshot of the MATLAB R2018a software interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. Below this is a toolbar with various icons. A red rectangle highlights the 'Run' button, which is represented by a right-pointing triangle icon. An arrow points from the 'Breakpoints' section towards the 'Run' button. Other visible buttons include 'New', 'Open', 'Save', 'Find Files', 'Compare', 'Go To', 'Find', 'Insert', 'Comment', 'Indent', 'fx', 'Breakpoints', 'Run Section', and a clock icon. The status bar at the bottom shows the current folder path: C:\Users\Edgard Montes Cortez\Documents\MATLAB\Inverted\_Pendulum.

- Implementación:

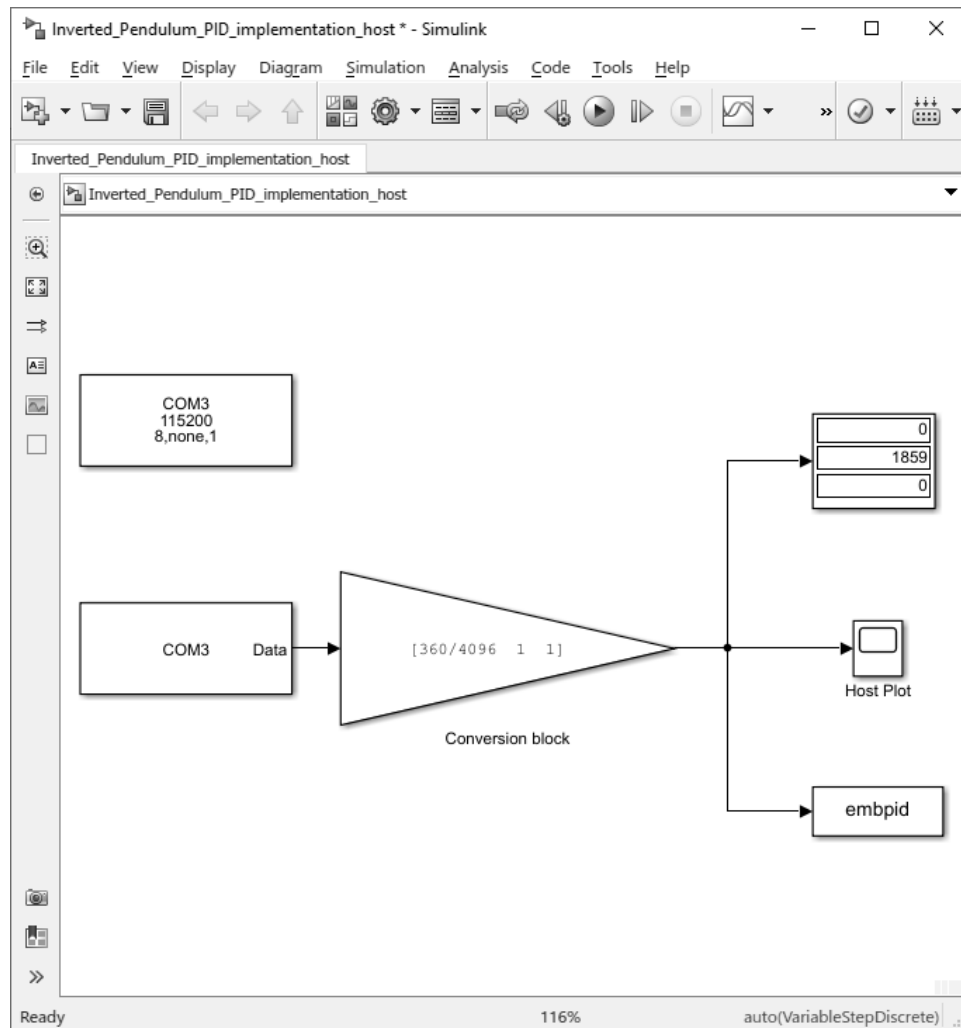
- Atornillar la varilla al eje del encoder sobre el carro quedando hacia abajo.
- Abrir el programa de Simulink *Inverted\_Pendulum\_PID\_implementation\_embedded.slx*.



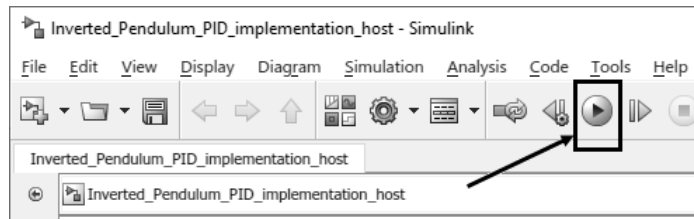
- Quemar el programa de simulink *Inverted\_Pendulum\_PID\_implementation\_embedded.slx* en el Arduino.



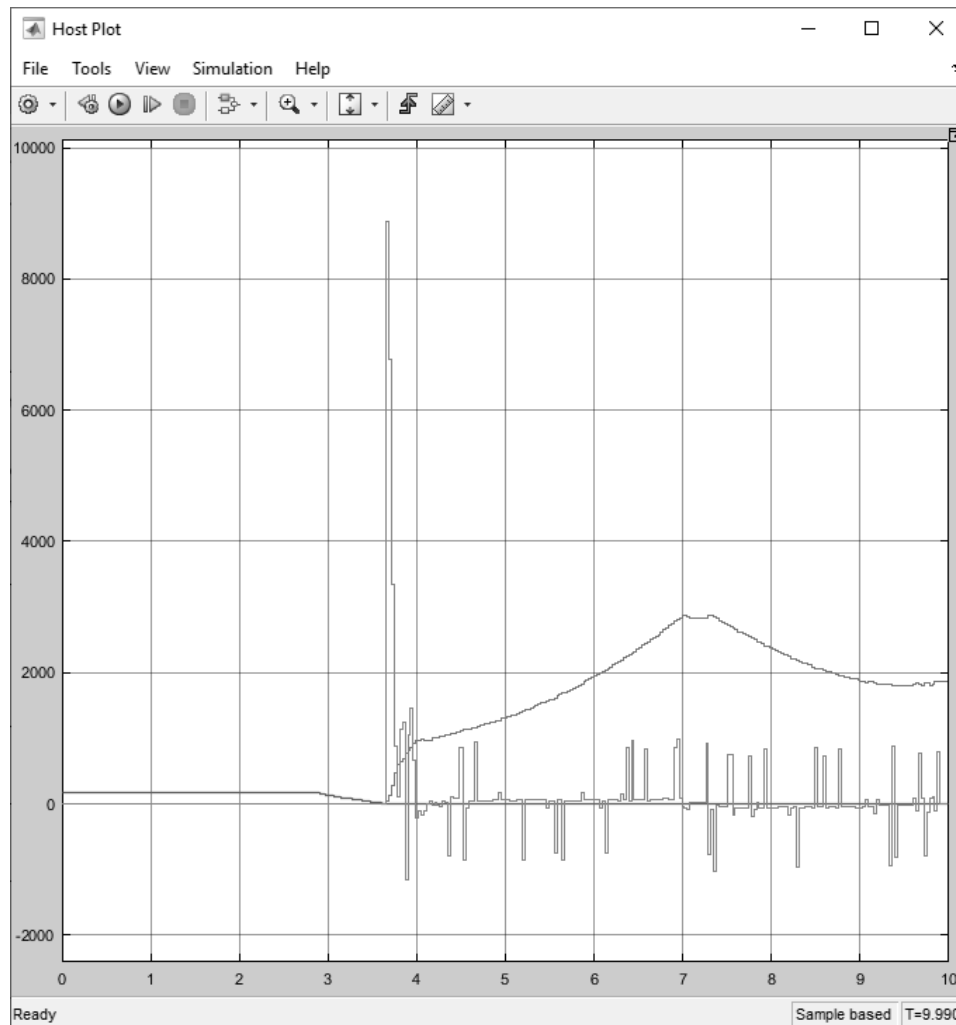
- Abrir el programa de Simulink *Inverted\_Pendulum\_PID\_implementation\_host.slx*.



- Verificar que en los bloques Serial Receive y Serial configuration esté seleccionado el puerto COM del Arduino.
- Dar click en el botón Run.



- Colocar la varilla en su posición vertical hacia arriba girándola en sentido contrario a las agujas del reloj, que es cuando el PID empieza a actuar.
- Dar doble click en el icono Host Plot para observar la gráfica del ángulo del péndulo, la posición del carro y la señal de control en tiempo real.



- Presionar en la tarjeta los botones + y – simultáneamente para mandar la señal de control a cero y poder colocar el péndulo hacia abajo.
- Estando el péndulo hacia abajo presionar el botón + o – para mover manualmente el carro.

- Estos datos se guardan en la variable `embpid` en el Workspace de Matlab para su posterior análisis.

## Diseño e implementación del controlador LQR

### Diseño:

- Abrir el script de Matlab *Inverted\_Pendulum\_LQR\_design.m*. El diseño del LQR está basado en el modelo del péndulo invertido sobre el carro, por lo que script llama a *Cart\_Pendulum\_identification.m* mediante la línea de comando: `run('Cart_Pendulum_identification.m');`
- El tiempo de muestreo `Tst` de transmisión de datos al igual que la zona muerta de motor están declarados en el script de Matlab *Inverted\_Pendulum\_LQR\_design.m*. Estas variables se utilizan posteriormente en los programas de Simulink *Inverted\_Pendulum\_LQR\_implementation\_embedded.m* e *Inverted\_Pendulum\_LQR\_implementation\_host.m*. Modificar las variables a conveniencia.

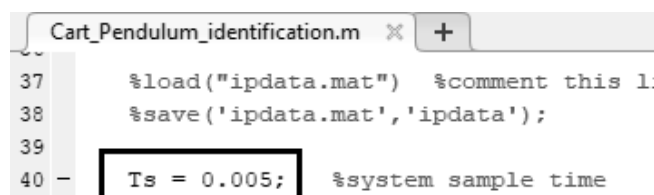


```

37
38 - Tst=0.03; %Implementation transmission sample time
39 - MotorDeadZone=35; %Implementation motor dead zone
40

```

- En el script *Cart\_Pendulum\_identification.m* modificar el tiempo de muestreo `Ts` del modelo discreto a utilizar en el diseño del LQR. También `Ts` es el tiempo de muestreo de todo el sistema en el programa de Simulink *Inverted\_Pendulum\_LQR\_implementation\_embedded.slx* que va embebido en el Arduino Mega 2560. De igual manera, comentar la línea de código `load('ipdata.mat');` si los datos a utilizar yacen del Workspace de Matlab.

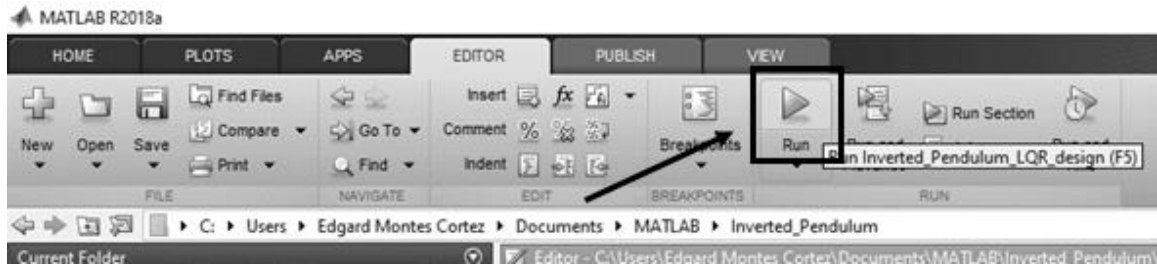


```

37 %load('ipdata.mat') %comment this l:
38 %save('ipdata.mat','ipdata');
39 Ts = 0.005;
40 - %system sample time

```

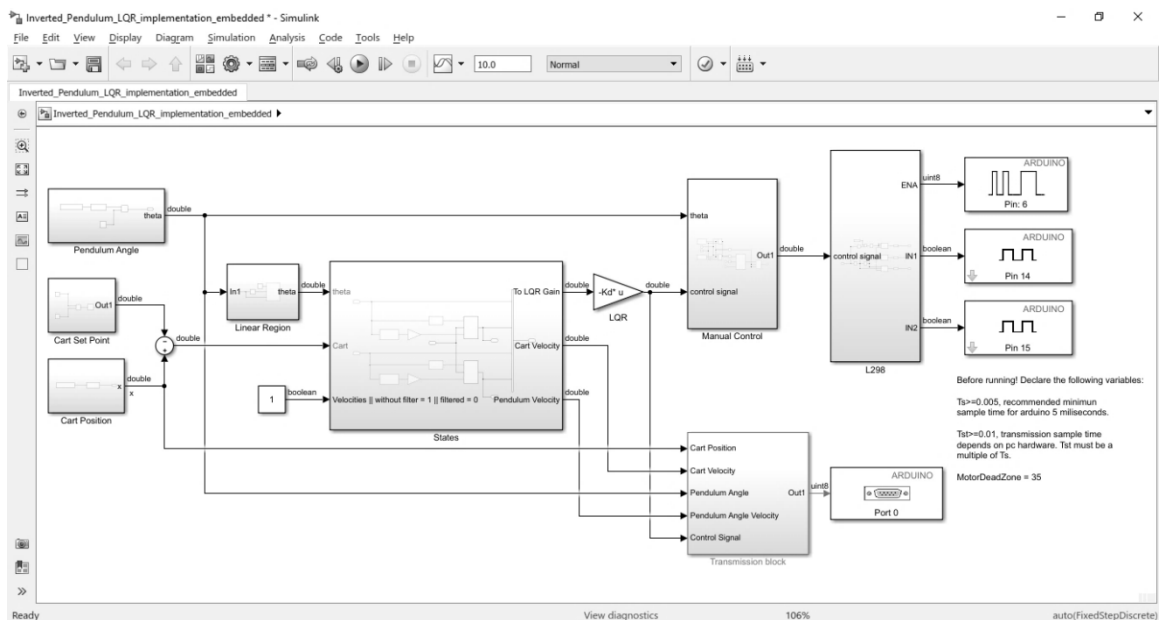
- En la pestaña Editor, dar click en Run o presionar F5 para ejecutar el script *Inverted\_Pendulum\_LQR\_design.m*



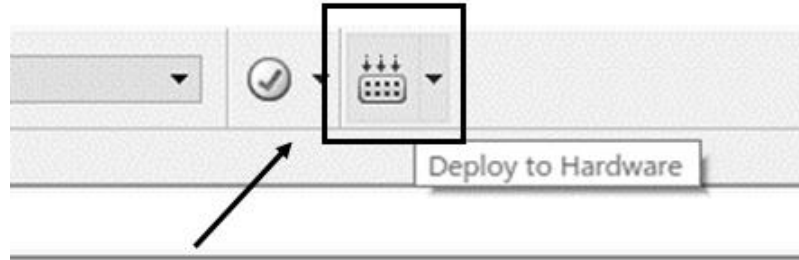
- El script da como resultado las figuras 59 y 68 de los acápites 2.8 y 2.9 respectivamente. En la ventana de comando se observa la controlabilidad y las ganancias óptimas para el sistema del péndulo invertido sobre el carro en tiempo continuo y discreto.

### Implementación:

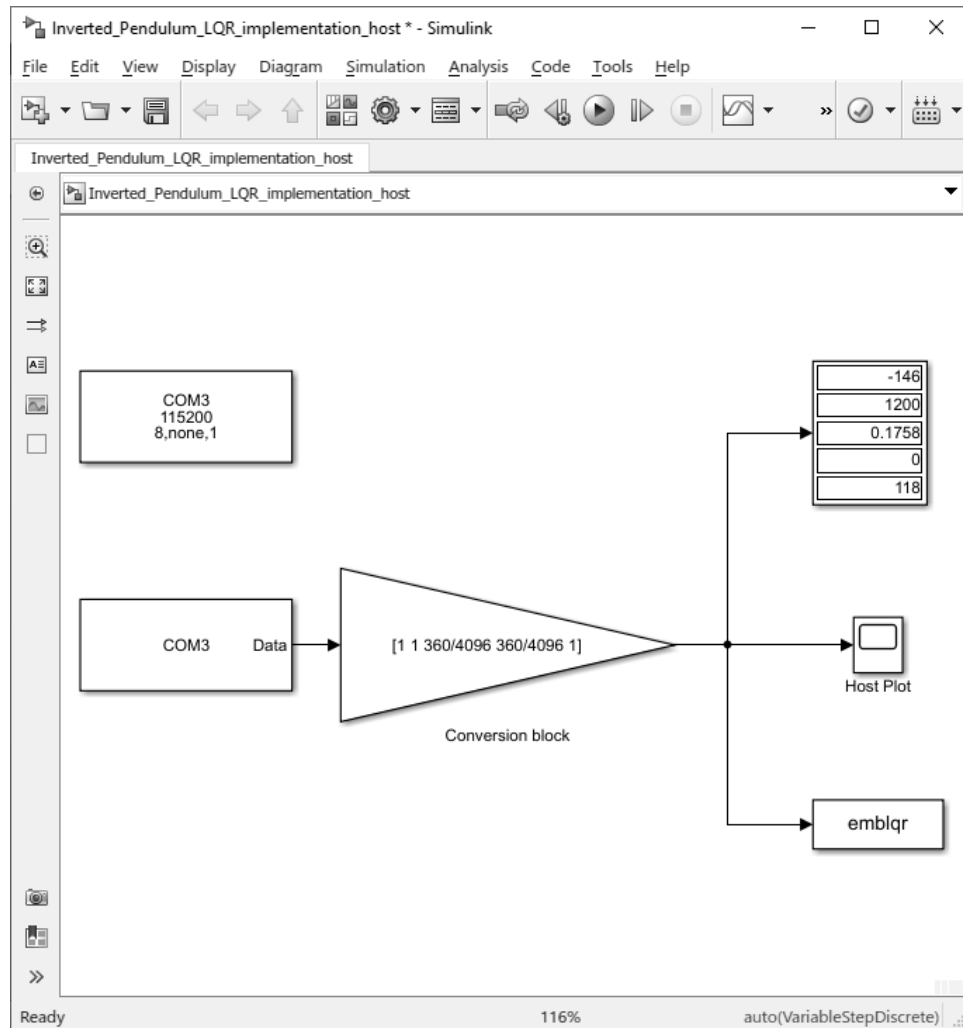
- Atornillar la varilla al eje del encoder sobre el carro quedando hacia abajo.
- Abrir el programa de Simulink *Inverted\_Pendulum\_LQR\_implementation\_embedded.slx*.



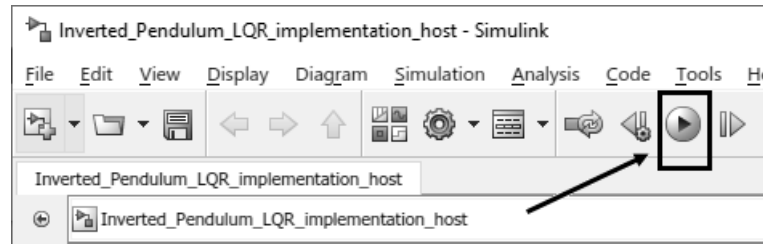
- Quemar el programa *Inverted\_Pendulum\_LQR\_implementation\_embedded.slx* en la tarjeta Arduino.



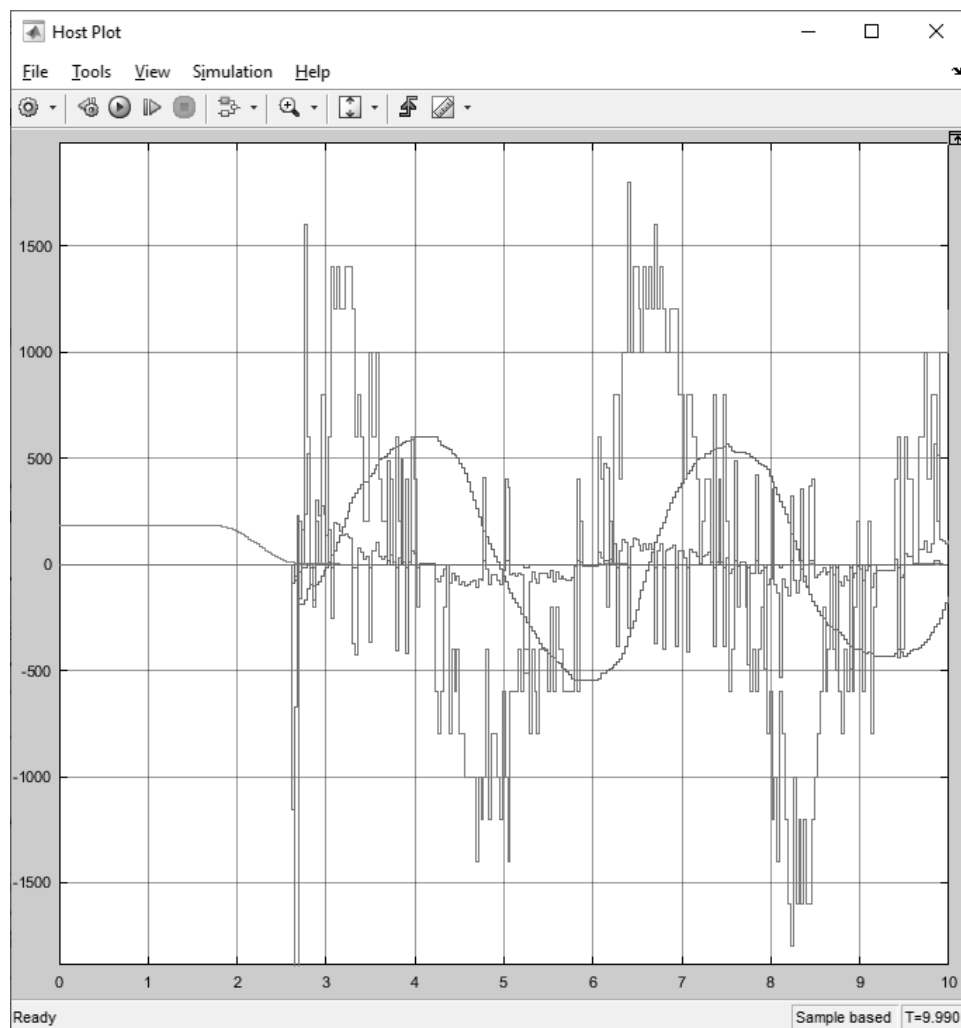
- Abrir el programa de Simulink *Inverted\_Pendulum\_LQR\_implementation\_host.slx*.



- Verificar que en los bloques Serial Receive y Serial configuration esté seleccionado el puerto COM del Arduino.
- Dar click en el botón Run.



- Colocar la varilla en su posición vertical hacia arriba girándola en sentido contrario a las agujas del reloj, que es cuando el LQR empieza a actuar.
- Dar doble click en el icono Host Plot para observar la gráfica de la posición y velocidad del carro y el péndulo, así como la señal de control.



- Presionar en la tarjeta los botones + y – simultáneamente para mandar la señal de control a cero y poder colocar el péndulo hacia abajo.
- Estando el péndulo hacia abajo presionar el botón + o – para mover manualmente el carro.
- Estos datos se guardan en la variable emblqr en el Workspace de Matlab para su posterior análisis.

### **Precauciones de seguridad**

- Alimentar el prototipo con un voltaje máximo de 12 VDC.
- Cuando no se vaya a utilizar el prototipo desconectarlo del tomacorriente.
- Si sobre el prototipo cae algún líquido u objeto, desconectarla de la red eléctrica y comprobar su funcionamiento con personal calificado.

### **Características técnicas del péndulo invertido**

#### Características eléctricas:

Voltaje de entrada del L298N	12 VDC
Voltaje de entrada de la tarjeta Arduino	9-12 VDC jack ó 5 VDC usb

#### Características físicas:

##### Dimensiones de la Impresora

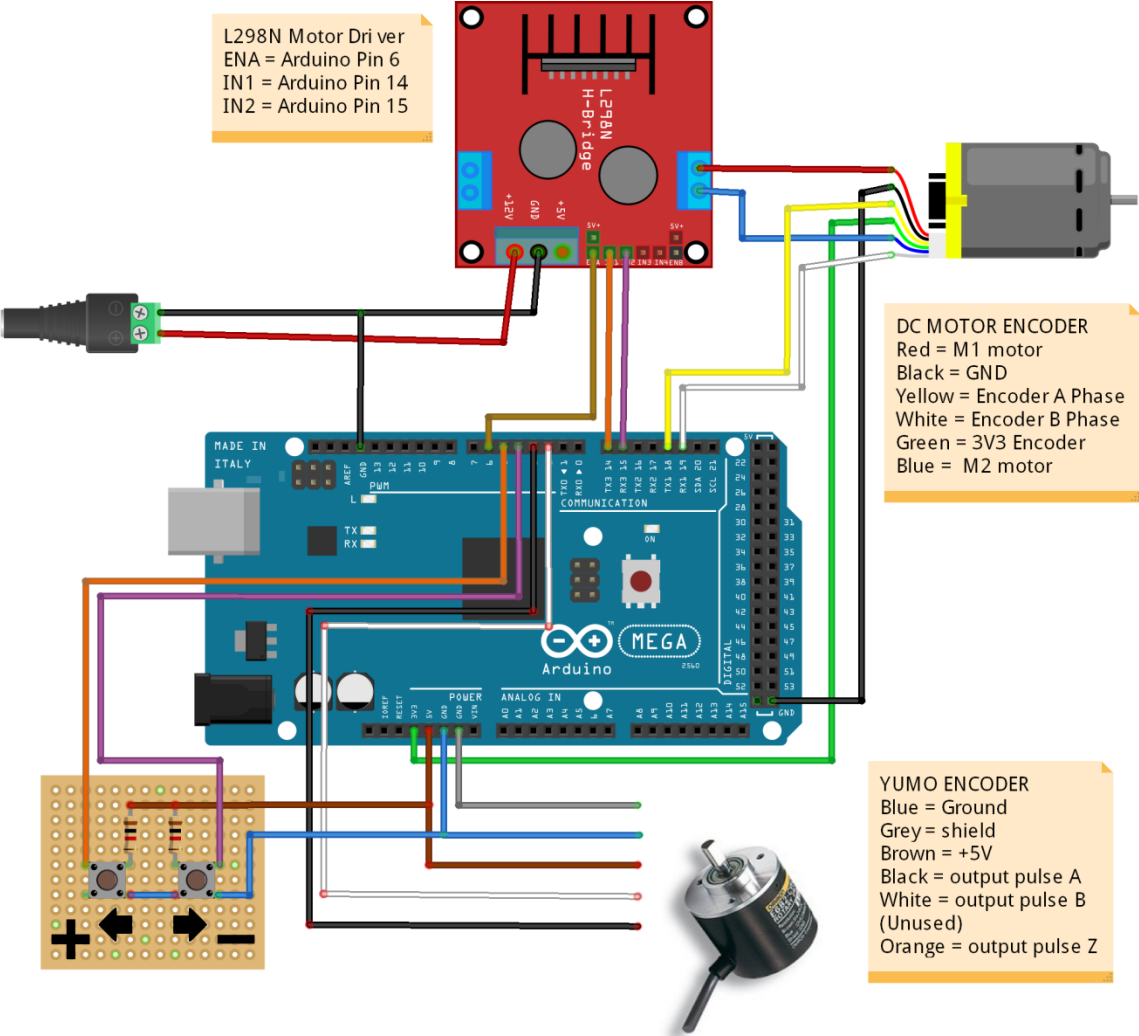
Largo	41 cm
Ancho	18 cm
Altura	17 cm

##### Dimensiones de la Varilla

Varilla no.1	87 cm
Varilla no. 2	17 cm
Peso total	6 lbs



Conexiones físicas



## Anexo 8. Lista de partes

**Tabla 1. Lista de partes.**

Componentes	Tienda	Precio US \$	Cantidad	Precio total US \$
Yumo Rotary Encoder 1024 P/R	www.sparkfun.com	39.95	1	39.95
12V 350RPM DC Gear Motor with Hall Encoder	www.dx.com	14.50	1	14.50
L298N motor driver	www.dx.com	3.60	2	7.20
Arduino Mega 2560 (Elegoo)	www.amazon.com	14.99	1	14.99
Tarjeta perforada	Ceca Managua	2.00	1	2.00
Push buttons 100 pcs	www.aliexpress.com	0.85	1	0.85
3 mts tubo de aluminio diá 8 mm	Alumicentro Managua	3.00	1	3.00
Jack dc 2.1x5 mm	Ceca Managua	1.50	1	1.50
Jumper hembra-hembra 40 pcs.	www.aliexpress.com	0.85	1	0.85
Impresora HP Deskjet 840C	-	gratis	1	gratis
			Total US \$	84.84

Fuente: Autor.

Nota: La impresora fue brindada por uno de los autores y estaba destinada a reciclaje.